Introducción a Redes Neuronales

Introduction

The Rosenblat Perceptro

The Perceptron
Learning Algorithr
Limitations

Combining Perceptrons

Gradient Based Learning

Loss Function

Fully Connected Networks

Introducción a Redes Neuronales

Dr. Mauricio Toledo-Acosta

Diplomado Ciencia de Datos con Python

Table of Contents

Introducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based

Loss Function

Fully Connected Networks

Introduction

- 2 The Rosenblatt Perceptron
 - The Perceptron Learning Algorithm
 - Limitations
 - Combining Perceptrons
- Gradient Based Learning
 - Loss Functions
 - Optimizers
- 4 Fully Connected Networks

Introduction

troducción Redes Neuronales

Introduction

The Rosenblat Perceptroi

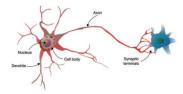
The Perceptron Learning Algorithm Limitations Combining

Gradient Based Learning Loss Function

Fully Connected Vetworks

Neural Networks

Artificial neural networks are machine learning techniques that simulate the mechanism of learning in biological organisms. The human nervous system contains cells, referred to as neurons. The neurons are connected to one another by of axons and dendrites, and the connecting regions between axons and dendrites are called synapses. The strengths of synaptic connections often change in response to external stimuli. This change is how learning takes place in living organisms.



El modelo lineal de clasificación

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based

Loss Function

Fully Connected Networks The Perceptrón algorithm (Rosenblatt, 1961) played an important role in Machine Learning history. It was first simulated in a computer IBM 704 at Cornell in 1957. By the early 60s, a dedicated hardware was designed to implement the learning algorithm.

El modelo lineal de clasificación

ntroducción a Redes Neuronales

Introduction

The
Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitation
Combining
Perceptrons

Gradient Based Learning Loss Function

Fully Connected Networks

- The Perceptrón algorithm (Rosenblatt, 1961) played an important role in Machine Learning history. It was first simulated in a computer IBM 704 at Cornell in 1957. By the early 60s, a dedicated hardware was designed to implement the learning algorithm.
- It was criticized by Marvin Minsky, who showed the limitations of the perceptron algorithm when dealing with a non linear separable set of points. This caused a void in the neural computation research lasting until the mid 80s.

Table of Contents

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations

Gradient Based

Ontimizers

Fully Connected Vetworks

- Introduction
- 2 The Rosenblatt Perceptron
 - The Perceptron Learning Algorithm
 - Limitations
 - Combining Perceptrons
- Gradient Based Learning
 - Loss Functions
 - Optimizers
- 4 Fully Connected Networks

The Rosenblatt Perceptron

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm

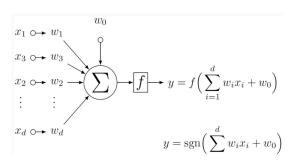
Combinions

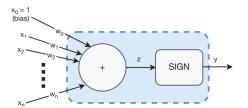
Combining Perceptrons

Gradient Based Learning

Ontimizers

Fully Connected Networks





ntroducción a Redes

Introduction

The Rosenblati Perceptroi

The Perceptron Learning Algorithm

Combining

Combining Perceptrons

Gradient Based Learning

Ontimizers

Fully Connected Networks We iterate over the training set

$$(x_1, y_1), (x_2, y_2) \cdots, (x_N, y_N).$$

Introducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm

Limitations

Combining Perceptrons

Gradient Based Learning

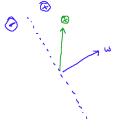
Ontimizers

Fully Connected Networks

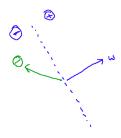
We iterate over the training set

$$(x_1, y_1), (x_2, y_2) \cdots, (x_N, y_N).$$

If the point is well classified:



(a) If
$$y = +1$$



(b) If
$$y = -1$$

ntroducción a Redes Neuronales

Introduction

Rosenblat Perceptro

The Perceptron Learning Algorithm

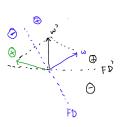
Combining

Combining Perceptrons

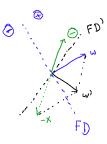
Based Learning

Loss Function Optimizers

Fully Connected Networks If the point is not well classified:



(a) If
$$y = +1$$



(b) If
$$y = -1$$

ntroducción a Redes Neuronales

Introduction

The Rosenblati Perceptroi

The Perceptron Learning Algorithm

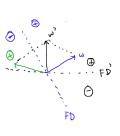
Limitations

Combining Perceptrons

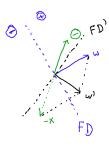
Gradient Based Learning

Loss Function Optimizers

Fully Connected Networks If the point is not well classified:



(a) If
$$y = +1$$



(b) If
$$y = -1$$

We update the vector w with w + yx

ntroducción a Redes Neuronales

ntroduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm

Combining

Perceptrons

Learning

Optimizers

Fully Connected Networks For each epoch, the algorithm is summarized as follows:

```
Data: A: training set of points, X = \{x_1, ..., x_N\} \subset \mathbb{R}^{D+1},
           Y = \{y_1, ..., y_N\}: set of labels
           \eta > 0: learning rate.
  Result: w: weight vector defining the decision frontier.
  Function Perceptron(X, y, \eta):
       w = 0:
       converged = False:
3
       while converged == False do
            for i \in \{1, ..., N\} do
                 if y_i \langle \mathbf{w}, x \rangle \leq 0 then
       return w:
```

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm

Combining

Gradient Based

Loss Functions

Fully Connected Networks For each epoch, the algorithm is summarized as follows:

```
Data: A: training set of points, X = \{x_1, ..., x_N\} \subset \mathbb{R}^{D+1},
        Y = \{y_1, ..., y_N\}: set of labels
        \eta > 0: learning rate.
Result: w: weight vector defining the decision frontier.
Function Perceptron(X, y, \eta):
     w = 0:
    converged = False;
     while converged == False do
          for i \in \{1, ..., N\} do
              if y_i \langle \mathbf{w}, x \rangle \leq 0 then
     return w:
```

w is a linear combination of $x_1, ..., x_N$.

Example

ntroducción a Redes

Introduction

The Rosenblatt Perceptron

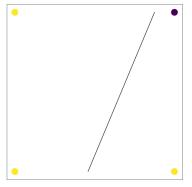
The Perceptron Learning Algorithm

Limitations

Combining Perceptrons

Gradient Based

Loss Functio



Example

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

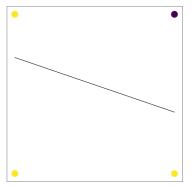
The Perceptron Learning Algorithm

Limitations

Combining Perceptrons

Based

Loss Function



Example

ntroducción a Redes

Introduction

The Rosenblatt Perceptror

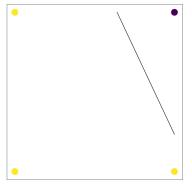
The Perceptron Learning Algorithm

Limitations

Combining Perceptrons

Based Learning

Loss Function



ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorith

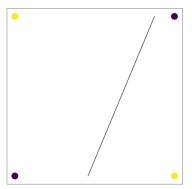
Limitations

Combining

Gradient Based

Ontimizers

Fully Connected Networks



ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorith

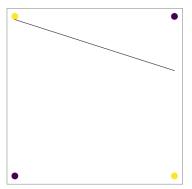
Limitations

Combining Perceptrons

Gradient Based

Ontimizers

Fully Connected Networks



ntroducción a Redes Neuronales

Introduction

Rosenblatt
Perceptror

The Perceptron Learning Algorith

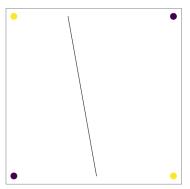
Limitations

Combining

Gradient Based

Ontimizers

Fully Connected Networks



ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorith

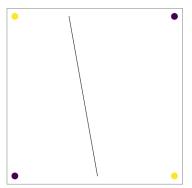
Limitations

Combining Perceptrons

Gradient Based

Ontimizers

Fully Connected Networks



ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorith

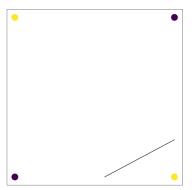
Limitations

Combining

Gradient Based

Ontimizers

Fully Connected Networks



ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorith

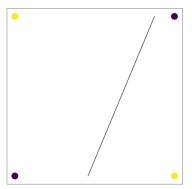
Limitations

Combining

Gradient Based

Ontimizers

Fully Connected Networks



ntroducción a Redes Neuronales

Introduction

The Rosenblat Perceptro

The Perceptron Learning Algorithm

Learning Algorithm
Limitations

Combining Perceptrons

Based Learning

Optimizers

Fully Connected Networks What happens if a straight line cannot separate the data points?



Alternatives:

- We change the model of a neuron or
- We combine multiple of them to solve this limitation.

Combining Perceptrons

ntroducción a Redes Neuronales

Introduction

The Rosenblati

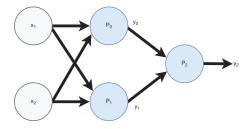
The Perceptron Learning Algorithm

Limitations

Combining Perceptrons

Gradient Based

Loss Functio



$$\begin{array}{c|cccc} -1 & (-1,-1) \\ +1 & (1,-1) \\ +1 & (-1,1) \\ -1 & (1,1) \end{array}$$

Combining Perceptrons

ntroducción a Redes Neuronales

Introduction

The Rosenblat

The Perceptron Learning Algorithm

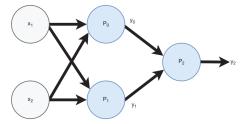
Combining

Perceptrons

Gradient Based Learning

Optimizers

Fully Connected Networks



This neural network is one of the simplest examples of a fully connected feedforward network. Fully connected means that the output of each neuron in one layer is connected to all neurons in the next layer. Feedforward means that there are no backward connections. A multilevel neural network has an input layer, one or more hidden layers, and an output layer. The input layer does not contain neurons but contains only the inputs themselves.

Table of Contents

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations Combining

Gradient Based Learning

Loss Functions

Fully Connected Networks Introduction

- 2 The Rosenblatt Perceptron
 - The Perceptron Learning Algorithm
 - Limitations
 - Combining Perceptrons
- Gradient Based Learning
 - Loss Functions
 - Optimizers
- 4 Fully Connected Networks

How to find the minimum of a function

Introducción a Redes

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Ontimizers

Fully Connectec Networks • Let f(x) be a function, we want to find the minimum of f.

How to find the minimum of a function

Introducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

Learning Algorithm
Limitations

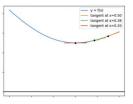
Combining

Gradient Based

Loss Function

Fully Connected Networks

- Let f(x) be a function, we want to find the minimum of f.
- The derivative at the point x that minimizes the value of f is 0.



How to find the minimum of a function

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Loss Function

Fully Connectec Networks

- Let f(x) be a function, we want to find the minimum of f.
- The derivative at the point x that minimizes the value of f is 0.
- Given an initial value x, the sign of the derivative f'(x) indicates in what direction to adjust x to reduce the value of f(x).

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Loss Functions

Fully Connected Networks

- Let f(x) be a function, we want to find the minimum of f.
- The derivative at the point x that minimizes the value of f is 0.
- Given an initial value x, the sign of the derivative f'(x) indicates in what direction to adjust x to reduce the value of f(x).
- Gradient descent uses the value of the derivative to decide how much to adjust x.

$$x_{n+1} = x_n - \eta f'(x_n).$$

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Loss Functions

Fully Connected Networks

- Let f(x) be a function, we want to find the minimum of f.
- The derivative at the point x that minimizes the value of f is 0.
- Given an initial value x, the sign of the derivative f'(x) indicates in what direction to adjust x to reduce the value of f(x).
- Gradient descent uses the value of the derivative to decide how much to adjust x.

$$x_{n+1} = x_n - \eta f'(x_n).$$

ullet If the learning rate η is to too large, gradient descent can overshoot the solution and fail to converge.

ntroducción a Redes Neuronales

Introduction

Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitations

Gradient Based Learning

Loss Functions

Fully Connected Networks

- Let f(x) be a function, we want to find the minimum of f.
- The derivative at the point x that minimizes the value of f is 0.
- Given an initial value x, the sign of the derivative f'(x) indicates in what direction to adjust x to reduce the value of f(x).
- Gradient descent uses the value of the derivative to decide how much to adjust x.

$$x_{n+1} = x_n - \eta f'(x_n).$$

- If the learning rate η is to too large, gradient descent can overshoot the solution and fail to converge.
- The algorithm is not guaranteed to find the global minimum because it can get stuck in a local minimum.

Learning Rate

Introducción a Redes

Introduction

The Rosenblatt Perceptron

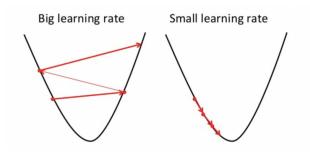
The Perceptron Learning Algorithm

Limitations

Combining

Gradient Based Learning

Loss Function



The high-dimensional case

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations

Gradient Based

Learning

Loss Function

Fully Connected Networks • Now, consider the function $f(x_0, x_1)$.

The high-dimensional case

Introducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

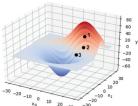
Combining

Gradient Based Learning

Loss Functions

Fully Connected Networks

- Now, consider the function $f(x_0, x_1)$.
- The gradient is a vector consisting of partial derivatives and indicates the direction in the input space that results in the steepest ascent for the value of f.



ntroduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

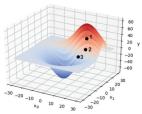
Combining Percentrons

Gradient Based Learning

Loss Function

Fully Connected Networks • Now, consider the function $f(x_0, x_1)$.

 The gradient is a vector consisting of partial derivatives and indicates the direction in the input space that results in the steepest ascent for the value of f.



• If we are at the point $\mathbf{x_n} = (x_0^{(n)}, x_1^{(n)})$ and want to minimize y, then we choose our next point as

$$\mathbf{x_{n+1}} = \mathbf{x_n} - \eta \nabla f(\mathbf{x_n})$$

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Loss Function

Fully Connected Networks When applying gradient descent to our neural network, we consider input values ${\bf x}$ to be constants, with our goal being to adjust the weights ${\bf w}$.

• Which function we want to minimize? A loss function.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based

Loss Functions

Fully Connected Networks When applying gradient descent to our neural network, we consider input values \mathbf{x} to be constants, with our goal being to adjust the weights \mathbf{w} .

- Which function we want to minimize? A loss function.
- In the context of an optimization algorithm, the function used to evaluate a candidate solution is referred to as the objective function. With neural networks, we seek to minimize the error. As such, the objective function is often referred to as a cost function or a loss function.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Combining Perceptrons

Gradient Based Learning

Loss Function

Fully Connected Networks In the case of the Perceptron, the loss function is given by

$$L^{(0/1)}(\mathbf{w}) = (y_i - \operatorname{sign}\langle \mathbf{w}, \mathbf{x}_i \rangle)^2$$

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron
Learning Algorithm
Limitations
Combining

Gradient Based Learning

Loss Function

Fully Connected Networks In the case of the Perceptron, the loss function is given by

$$L^{(0/1)}(\mathbf{w}) = \frac{1}{2}(y_i - \operatorname{sign}\langle \mathbf{w}, \mathbf{x}_i \rangle)^2 = 1 - y_i \operatorname{sign}\langle \mathbf{w}, \mathbf{x}_i \rangle$$

 This function is not smooth, we use the smooth surrogate loss function

$$L(\mathbf{w}) = \max\{-y_i \operatorname{sign}\langle \mathbf{w}, \mathbf{x}_i \rangle, 0\}.$$

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Loss Function

Fully Connected Networks • In the case of the Perceptron, the loss function is given by

 This function is not smooth, we use the smooth surrogate loss function

$$L(\mathbf{w}) = \max\{-y_i \operatorname{sign}\langle \mathbf{w}, \mathbf{x}_i \rangle, 0\}.$$

Applying gradient descent

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \nabla L(\mathbf{w})$$

$$= \begin{cases} \mathbf{w}_n + \eta y_i \mathbf{x}_i, & \text{well classified} \\ \mathbf{w}_n, & \text{misclassified} \end{cases}$$

Loss Functions

ntroducción a Redes

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Combining Percentrons

Gradient Based Learning

Loss Functions

Fully Connected

Other examples of loss functions

▶ Keras Losses

Loss Functions: MSE

ntroducción a Redes Neuronales

Introduction

Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Limitations
Combining
Perceptrons

Gradient Based Learning

Loss Functions

Fully Connected Networks

Mean Square Error (MSE)

$$L(y,t)=(t-y)^2.$$

L2 loss.

Loss Functions: MSE

ntroducción a Redes Neuronales

Introductio

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Limitations
Combining
Perceptrons

Gradient Based Learning

Loss Functions

Fully Connected Networks

Mean Square Error (MSE)

$$L(y,t)=(t-y)^2.$$

- L2 loss.
- Good for regression tasks.

Loss Functions: MSE

ntroducción a Redes Neuronales

Introductio

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations

Gradient Based

Loss Functions

Optimizers

Fully Connectec Networks

Mean Square Error (MSE)

$$L(y,t)=(t-y)^2.$$

- L2 loss.
- Good for regression tasks.
- Trivial derivative for gradient descent.

Loss Functions: MAE

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Limitations
Combining

Gradient Based Learning

Loss Functions

Loss Functio

Fully Connected Networks

Mean Absolute Error (MAE)

$$L=|t-y|.$$

- L1 loss.
- More robust to outliers than mse.
- Good for regression tasks.
- Discontinuity in its derivative.

Loss Functions: MAE

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Limitations
Combining

Gradient Based Learning

Loss Functions

Loss Functio

Fully Connected Networks

Mean Absolute Error (MAE)

$$L=|t-y|.$$

- L1 loss.
- More robust to outliers than mse.
- Good for regression tasks.
- Discontinuity in its derivative.

Loss Functions: Hinge

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations

Gradient Based

Loss Functions

Fully Connected Networks

Hinge Loss

$$L = \max\{-y_i\hat{y}_i, 0\}.$$

- Used in SVMs and Perceptron.
- Penalizes errors, but also correct predictions of low confidence (probabilities).
- Good for binary classification tasks.

Loss Functions: Categorical cross entropy

ntroducción a Redes Neuronales

Introduction

The Rosenblat Perceptro

The Perceptron Learning Algorithm

Limitations

Combining

Gradient

Based Learning

Loss Functions

Fully Connected Networks

Categorical cross entropy

$$L = \sum_{i}^{n} y_{i} \log (\hat{y}_{i}).$$

- Good for multi-class classification problems.
- Considers y to be a one-hot encoding vector in n classes.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Combining Perceptrons

Based Learning

Optimizers

Fully Connected Networks Gradient descent is the most basic but most used optimization algorithm.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based

Loss Functio
Optimizers

Fully Connected Networks Gradient descent is the most basic but most used optimization algorithm. It's used heavily in linear regression and classification algorithms.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations

Combining Perceptron

Gradient Based Learning

Optimizers

Fully Connected Networks Gradient descent is the most basic but most used optimization algorithm. It's used heavily in linear regression and classification algorithms. Gradient descent is a first-order optimization algorithm which is dependent on the first order derivative of a loss function.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Combining Perceptrons

Based Learning

Optimizers

Fully Connected Networks

Advantages:

- Easy computation.
- Easy to implement.
- Easy to understand.

Introducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Combining Perceptro

Gradient Based Learning Loss Function

Optimizers

Fully Connected Vetworks

Advantages:

- Easy computation.
- Easy to implement.
- Easy to understand.

Disadvantages:

- May trap at local minima.
- Weights are changed after calculating gradient on the whole dataset. May take a long time to converge.
- Requires large memory to calculate gradient on the whole dataset.

Optimizers

ntroducción a Redes Neuronales

Introductio

The Rosenblati Perceptroi

The Perceptron Learning Algorithm Limitations

Limitations
Combining

Gradient Based Learning

Optimizers

Fully Connected Networks

- Gradient Descent
- Stochastic Gradient Descent
- Stochastic Gradient descent with momentum
- Mini-Batch Gradient Descent
- Adagrad
- RMSProp
- AdaDelta
- Adam

→ Keras Optimizers

ntroducción a Redes Neuronales

Introduction

The Rosenblatt

The Perceptron Learning Algorithm Limitations

Combining Perceptrons

Based Learning

Loss Function
Optimizers

Fully Connected Networks • Instead of taking the whole dataset for each iteration in each iteration, we randomly shuffle the data and take a batch.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Optimizers

- Instead of taking the whole dataset for each iteration in each iteration, we randomly shuffle the data and take a batch.
- The path took by the algorithm is full of noise as compared to the gradient descent algorithm.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations Combining

Gradient Based Learning

Optimizers

- Instead of taking the whole dataset for each iteration in each iteration, we randomly shuffle the data and take a batch.
- The path took by the algorithm is full of noise as compared to the gradient descent algorithm.
- Due to an increase in the number of iterations, the overall computation time increases. Still, the computation cost is still less than that of the gradient descent optimizer.

ntroducción a Redes Neuronales

Introduction

Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitations

Gradient
Based
Learning
Loss Function
Ontimizers

Fully Connected • Instead of taking the whole dataset for each iteration in each iteration, we randomly shuffle the data and take a batch.

- The path took by the algorithm is full of noise as compared to the gradient descent algorithm.
- Due to an increase in the number of iterations, the overall computation time increases. Still, the computation cost is still less than that of the gradient descent optimizer.
- If the data is enormous and computational time is an essential factor, stochastic gradient descent should be preferred over batch gradient descent algorithm.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm

Limitations

Combining Perceptrons

Based Learning

Optimizers

Fully Connectec Networks The adaptive gradient descent algorithm uses different learning rates for each iteration. The change in learning rate depends upon the difference in the parameters during training.

ntroducción a Redes Neuronales

Introduction

The Rescention

The Perceptron Learning Algorithm Limitations Combining Perceptrons

Gradient Based Learning

Optimizers

- The adaptive gradient descent algorithm uses different learning rates for each iteration. The change in learning rate depends upon the difference in the parameters during training.
- The more the weights change, the least the learning rate changes.

Introducción a Redes Neuronales

Introduction

The
Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitations
Combining
Perceptrons

Gradient Based Learning Loss Function

Optimizers

- The adaptive gradient descent algorithm uses different learning rates for each iteration. The change in learning rate depends upon the difference in the parameters during training.
- The more the weights change, the least the learning rate changes.
- The benefit of using Adagrad is that it abolishes the need to modify the learning rate manually. It is more reliable than gradient descent algorithmss, as it reaches convergence at a higher speed.

ntroducción a Redes Neuronales

Introductio

The
Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitations
Combining
Perceptrons

Gradient
Based
Learning
Loss Function
Ontimizers

Fully Connected The adaptive gradient descent algorithm uses different learning rates for each iteration. The change in learning rate depends upon the difference in the parameters during training.

- The more the weights change, the least the learning rate changes.
- The benefit of using Adagrad is that it abolishes the need to modify the learning rate manually. It is more reliable than gradient descent algorithmss, as it reaches convergence at a higher speed.
- One downside of AdaGrad optimizer is that it decreases the learning rate aggressively and monotonically. There might be a point when the learning rate becomes extremely small.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt

The Perceptron Learning Algorithm Limitations

Combining Perceptrons

Based Learning

Optimizers

Fully Connected Networks • It is an extension of stochastic gradient descent.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Combining

rerceptrons

Based Learning

Optimizers

- It is an extension of stochastic gradient descent.
- Adam optimizer updates the learning rate for each network weight individually.

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron
Learning Algorithm
Limitations

Gradient Based

Loss Functio

Fully Connected

- It is an extension of stochastic gradient descent.
- Adam optimizer updates the learning rate for each network weight individually.
- The Adam optimizers inherit the features of both Adagrad and RMSProp algorithms.

ntroducción a Redes Neuronales

Introduction

The
Rosenblatt
Perceptron
The Perceptron
Learning Algorith
Limitations

Gradient Based Learning

Optimizers

- It is an extension of stochastic gradient descent.
- Adam optimizer updates the learning rate for each network weight individually.
- The Adam optimizers inherit the features of both Adagrad and RMSProp algorithms.
- Instead of adapting learning rates based upon the first moment (mean), it also uses the second moment of the gradients (variance).

ntroducción a Redes Neuronales

Introduction

I he
Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitations
Combining

Gradient
Based
Learning
Loss Function
Ontimizers

Fully Connected Networks • It is an extension of stochastic gradient descent.

- Adam optimizer updates the learning rate for each network weight individually.
- The Adam optimizers inherit the features of both Adagrad and RMSProp algorithms.
- Instead of adapting learning rates based upon the first moment (mean), it also uses the second moment of the gradients (variance).
- It is often used as a default optimization algorithm. It has faster running time, low memory requirements, and requires less tuning than any other optimization algorithm.

ntroducción a Redes Neuronales

Introductio

The
Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitations
Combining
Perceptrons

Gradient
Based
Learning
Loss Function
Optimizers

- It is an extension of stochastic gradient descent.
- Adam optimizer updates the learning rate for each network weight individually.
- The Adam optimizers inherit the features of both Adagrad and RMSProp algorithms.
- Instead of adapting learning rates based upon the first moment (mean), it also uses the second moment of the gradients (variance).
- It is often used as a default optimization algorithm. It has faster running time, low memory requirements, and requires less tuning than any other optimization algorithm.
- It tends to focus on faster computation time, it might not generalize the data well enough.

Optimizers

ntroducción a Redes

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorith

Limitations

Combining

Gradient Based

Learning

Optimizers

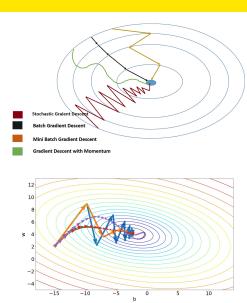


Table of Contents

Introducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient

Loss Function

- Introduction
- 2 The Rosenblatt Perceptron
 - The Perceptron Learning Algorithm
 - Limitations
 - Combining Perceptrons
- Gradient Based Learning
 - Loss Functions
 - Optimizers
- Fully Connected Networks

Fully Connected Networks

ntroducción a Redes Neuronales

Introductio

The Rosenblat Perceptron

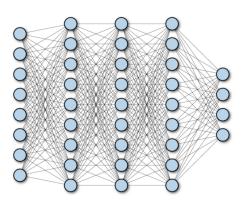
The Perceptron Learning Algorithm

Combining

Based Learning

Optimizers

Fully Connected Networks



A fully connected neural network consists of a series of fully connected layers that connect every neuron in one layer to every neuron in the other layer.

Advantages and Disadvantages

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Combining Percentrops

Gradient Based Learning

Optimizers

Fully Connected The major advantage of fully connected networks is that they are no special assumptions needed to be made about the input.

Advantages and Disadvantages

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Gradient Based Learning Loss Functio

Fully Connected

- The major advantage of fully connected networks is that they are no special assumptions needed to be made about the input.
- While being structure agnostic makes fully connected networks very broadly applicable, such networks do tend to have weaker performance than special-purpose networks tuned to the structure of a problem space.

The Algorithm

ntroducción a Redes Neuronales

Introductio

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations

Gradient Based Learning

Loss Functio

Fully Connected

The algorithm consists of three simple steps:

• First, present one or more training examples to the neural network: Feed-Forward.

The Algorithm

ntroducción a Redes Neuronales

Introductio

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations Combining

Gradient Based Learning

Optimizers

Fully Connected Networks

The algorithm consists of three simple steps:

- First, present one or more training examples to the neural network: Feed-Forward.
- Second, compare the output of the neural network to the desired value: Loss function.

The Algorithm

ntroducción a Redes Neuronales

Introductio

Rosenblatt
Perceptron
The Perceptron
Learning Algorithm
Limitations
Combining

Gradient
Based
Learning
Loss Function
Optimizers

Fully Connected Networks The algorithm consists of three simple steps:

- First, present one or more training examples to the neural network: Feed-Forward.
- Second, compare the output of the neural network to the desired value: Loss function.
- Finally, adjust the weights to make the output get closer to the desired value using gradient descent: Back propagation.

Feed-Forward

ntroducción a Redes Neuronales

Introduction

The Rosenblat Perceptro

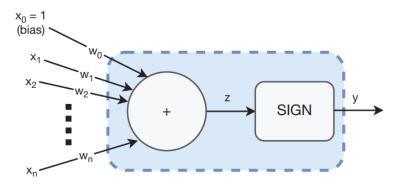
The Perceptron Learning Algorithm Limitations

Combining Percentrons

Based Learning

Optimizers

Fully Connected Networks



The sign function is an activation function of the neuron. It is not smooth because of the discontinuity in 0.

Other Activation Functions

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm

Limitations

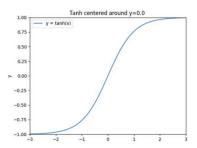
Combining

Based Learning

Loss Functio

Fully Connected Networks

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$



$$\tanh'(x) = 1 - \tanh^2(x)$$

Other Activation Functions

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm

Limitations

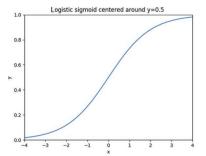
Combining

Based Learning

Loss Functio

Fully Connected

$$S(x) = \frac{e^x}{e^x + 1}$$



$$S'(x) = S(x)(1 - S(x))$$

Choice of Activation Functions

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorithm Limitations

Combining

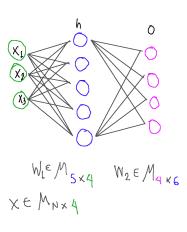
Gradient Based Learning

Loss Functio

Fully Connected There exist a large number of activation functions. Two popular choices are tanh and the logistic sigmoid function. When picking between the two, choose tanh for hidden layers and logistic sigmoid for the output layer.

► Keras Activation Functions

Feed-Forward: An Example



$$X = \begin{pmatrix} 1 & X_{1}^{(1)} & X_{2}^{(1)} & X_{3}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{1}^{(N)} & X_{2}^{(N)} & X_{3}^{(N)} \end{pmatrix}$$

$$M = \bigoplus_{1} \begin{pmatrix} X \cdot W_{1}^{T} \end{pmatrix}$$

$$O = \bigoplus_{2} \begin{pmatrix} \overline{h} \cdot W_{2}^{T} \end{pmatrix}$$

$$h \in M$$

$$N \times 5$$

$$\overline{h} \in M$$

Example: Back-propagation

ntroducción a Redes Neuronales

Introduction

The Rosenblat Perceptro

The Perceptron Learning Algorithm

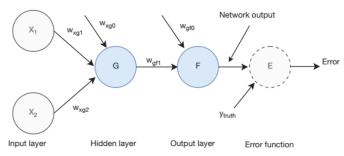
Limitations

Perceptro

Based Learning

Optimizers

Fully Connected



This neural network implements

$$\hat{y} = S(w_{gf0} + w_{gf1} \tanh(w_{xg0} + w_{xg1}x_1 + w_{xg2}x_2))$$

We use the MSE:

$$e(f) = \frac{(y-f)^2}{2}$$



Fully Connected Networks • The error is given by

$$\mathsf{Error} = \frac{1}{2} \left(y - S \left(w_{gf0} + w_{gf1} \mathsf{tanh} \left(w_{\mathsf{x}g0} + w_{\mathsf{x}g1} \mathsf{x}_1 + w_{\mathsf{x}g2} \mathsf{x}_2 \right) \right) \right.$$

We write it as:

$$e(f) = \frac{1}{2}(y - f)^{2}$$

$$f(z_{f}) = S(z_{f})$$

$$z_{f}(w_{gf0}, w_{gf1}, g) = w_{gf0} + w_{gf1}g$$

$$g(z_{g}) = \tanh(z_{g})$$

$$z_{g}(w_{xg0}, w_{xg1}, w_{xg2}) = w_{xg0} + w_{xg1}x_{1} + w_{xg2}x_{2}$$

Example: Back-propagation

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptron

The Perceptron Learning Algorithm Limitations

Limitations

Combining Perceptrons

Based Learning

Loss Functio

Fully Connected Networks

$$\begin{split} &\frac{\partial e}{\partial w_{gf0}} = -(y-f) \cdot S'(z_f) \\ &\frac{\partial e}{\partial w_{gf1}} = -(y-f) \cdot S'(z_f) \cdot g \\ &\frac{\partial e}{\partial w_{xg0}} = -(y-f) \cdot S'(z_f) \cdot w_{gf1} \cdot tanh'(z_g) \\ &\frac{\partial e}{\partial w_{xg1}} = -(y-f) \cdot S'(z_f) \cdot w_{gf1} \cdot tanh'(z_g) \cdot x_1 \\ &\frac{\partial e}{\partial w_{xg2}} = -(y-f) \cdot S'(z_f) \cdot w_{gf1} \cdot tanh'(z_g) \cdot x_2 \end{split}$$

Example: Back-propagation

ntroducción a Redes Neuronales

Introduction

The Rosenblatt Perceptror

The Perceptron Learning Algorithm

Limitations Combining

Gradient Based Learning

Loss Functio Optimizers

Fully Connected Networks Finally, we update the weights, via gradient descent

$$\begin{aligned} w_{gf0} &\leftarrow w_{gf0} - \eta \frac{\partial e}{\partial w_{gf0}} \\ w_{gf1} &\leftarrow w_{gf1} - \eta \frac{\partial e}{\partial w_{gf1}} \\ w_{xg0} &\leftarrow w_{xg0} - \eta \frac{\partial e}{\partial w_{xg0}} \\ w_{xg1} &\leftarrow w_{xg1} - \eta \frac{\partial e}{\partial w_{xg1}} \\ w_{xg2} &\leftarrow w_{xg2} - \eta \frac{\partial e}{\partial w_{xg2}} \end{aligned}$$