Universidade Federal do Espírito Santo
Centro de Ciências Agrárias – CCA UFES
Departamento de Computação

Otimização por Descida de Gradiente

Redes Neurais Artificiais

Site: http://jeiks.net
E-mail: jacsonrcsilva@gmail.com

Otimização por Descida de Gradiente

- Este é um método iterativo;
- Porém, não possui garantia de encontrar parâmetros ótimos,

  a não ser que o problema seja convexo;
- Esta técnica utiliza somente derivadas de 1ª ordem (∇E de E).
  - Gradiente: vetor composto pelas derivadas parciais;
  - Matriz Jacobiana.

Princípio de Aplicação

- É aplicado em um problema de aprendizagem supervisionada com mapeamento funcional g(x;w).
- Deve-se definir uma função de erro **E(w)** dos parâmetros livres w.
- A forma analítica do **E** é desconhecida;
- O valor do E para um argumento concreto E(τ) é calculável;
- É diferenciável em relação aos w = (w0, …, wi, ..., wd)^T:

$$\frac{d\,E(\underline{w})}{d\,\underline{w}} = grad\,E(\underline{w}) = \nabla E(\underline{w})$$

- O valor de ∇E(w(τ)) para o argumento concreto w(τ) é calculável.

Exemplo

- Abra o Octave e crie uma função para retornar o resultado de:

$$f(x) = (x - 4.5 - \cos(2\,(x - 4.5)))^2 + 2\,(x - 5) + 4$$

- Crie um vetor *x* dentre *9,5* à *11,5*, com intervalos de *0,1*.
- Plote o gráfico dessa função.
- Crie um ponto x1 = 11,4.
  - Obtenha o valor desse x1 na função;
  - Plote-o no gráfico: plot(x1 , funcao(x1), 'r*')
- Tente o mesmo com x1 = 9,6.
- Baseando-se no gradiente, encontre o mínimo da função:
  - Com um ponto, calcule a derivada: f '(x) = Δy/Δx → “f '(x)< 0” ou “ f '(x) > 0”;
  - Agora, vá na direção oposta do gradiente;
  - Princípio da descida do gradiente:

$$y(1) = y(0) - \eta f'(x), \eta > 0$$
$$w(1) = w(0) + \Delta w(0) = w(0) + \eta(-\nabla E(w))$$

Descida do Gradiente

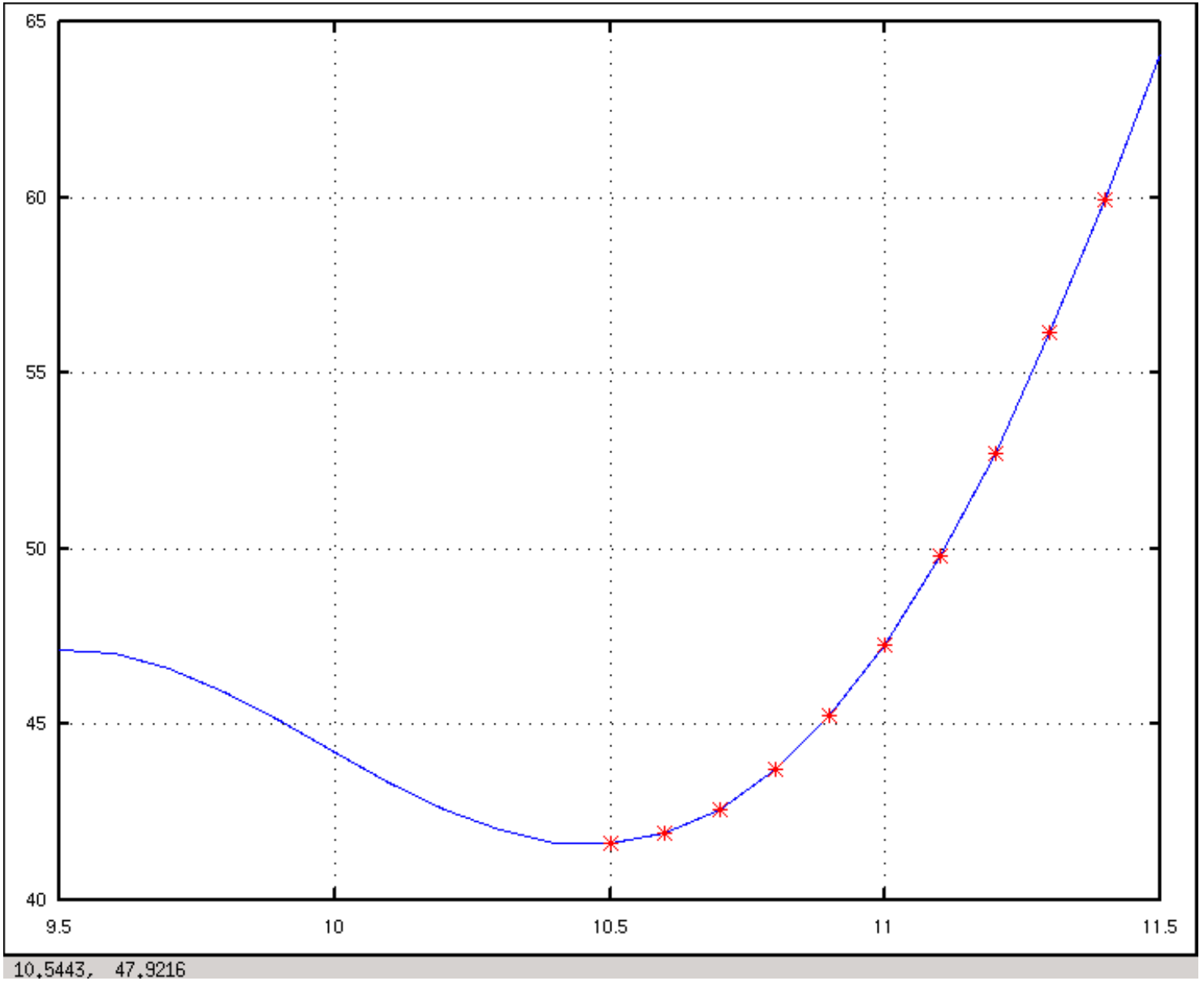
- Fato:
  - E(w) cresce na direção positiva do gradiente.

    "*em vizinhança próxima ao valor concreto w(τ)*"
- Estratégia:
  - Assim, E(w) decresce na direção negativa (oposta) do gradiente.
  - Então, o valor de w(τ) é modificado iterativamente nessa direção, usando um múltiplo do gradiente negativo.

$$w(\tau+1) = w(\tau) + \eta[-\nabla E(w(\tau))]$$
$$\eta = \text{taxa de aprendizagem}$$
$$\underline{w}(0) = \text{valor inicial}$$

- Efeitos:
  - Convergência de E(w) para um mínimo;
  - Mínimo atingido não necessariamente é o mínimo global.
  - η pequena ⇒ aprendizagem lenta
  - η grande ⇒ oscilações, divergências.

Encontraram o mínimo da função?



Tentem plotar a mesma função com:

X = -20:0.1:20;

Agora encontre seu mínimo!

Exemplos

- Supondo

  o conhecimento analítico de E(w) e

  $\eta = 0{,}2$

- Encontre o mínimo para:

  a) $E(w) = w^2$,

  com $w(0) = 3$

  b) $E(w) = 0{,}034w^4 - 0{,}02w^3 - 0{,}12w^2 + 0{,}35w + 4$,

  com $w(0) = -6$

- Após isso, plote as funções e os pontos de iteração no gráfico gerado.

Encontrando o argumento mínimo

1-D

- Busca-se o x* = arg min f(x)
- Não tem-se a função, mas possui-se o valor de entrada (x_i) e o resultado (y_i=f(x_i)) para encontrar o mínimo.
- Escolhe-se então um valor aleatório de x_i e obtém-se seu resultado.
- Então, obtém-se sua derivada (que é seu gradiente) e caminha-se sentido contrário.
- Lembrando-se que a função em 1-D é uma reta.

$$x_{i+1} = x_i + \eta\left(-f'(x_i)\right) \Rightarrow f(x_{i+1}) < f(x_i)$$

$$|x_{i+1} - x_i| \leq \varepsilon \text{ pequeno}$$

Encontrando o argumento mínimo

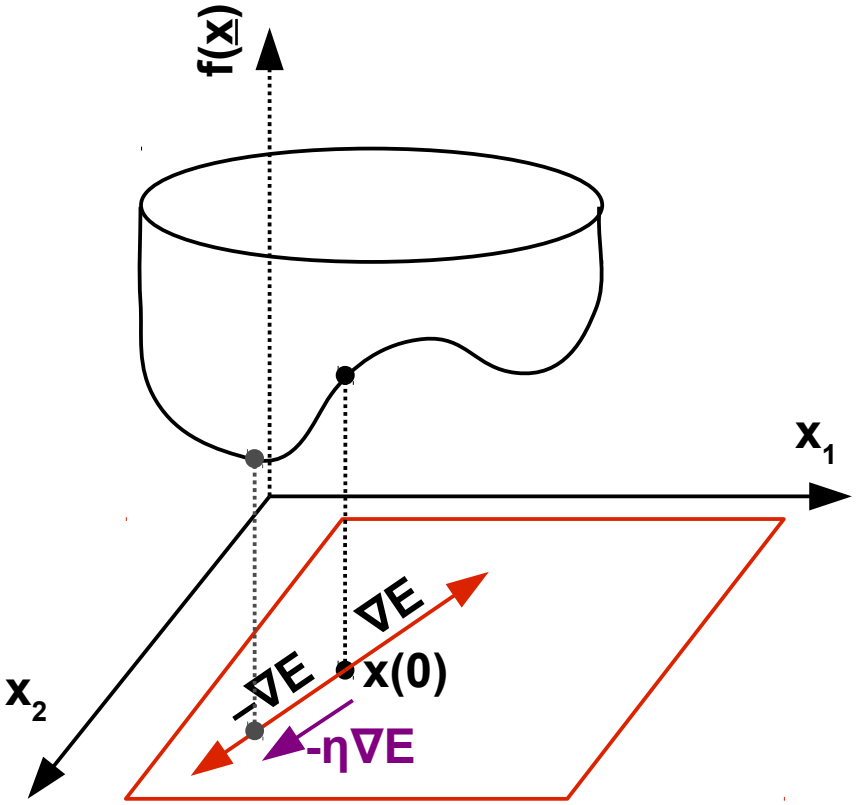
2-D

- Busca-se

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = argmin\, f \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

- Seu gradiente é definido por

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{pmatrix}$$



- Lembrando-se que a função em 2-D é um plano.
- Caminha-se no sentido contrário de ∇E.

Atividade

- Encontre o argumento mínimo em:

$$f\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = f\begin{pmatrix} -3\,x_1 \\ 1+4\,x_2 \end{pmatrix} = -3\,x_1 + 1 + 4\,x_2$$

- Sendo:

$$\underline{w}(\tau+1) = \underline{w}(\tau) + \eta\left(-\nabla E(\underline{w})\right)$$

$$\underline{w}_j(\tau+1) = \underline{w}(\tau) - \eta\left(\frac{\partial E(\underline{w})}{\partial w_j(\tau)}\right), j = 1, \ldots, d$$

$$RSS(\underline{w}) = E(\underline{w}) = (\hat{y} - y)^T(\hat{y} - y) = \|\hat{y} - y\|^2$$

$$\hat{y}^{(k)}(\underline{w}) = \underline{w} \cdot \underline{x}^{(k)}$$

Aprendizagem supervisionada com descida de gradiente

- Em problemas lineares:

$$\hat{y}_i(\underline{x};\underline{w}) = \underline{w}_i \cdot \underline{x};$$

- A definição da função de erro é:

$$E(w) = E[\|\hat{y}(w) - y\|^2]$$

- Aproximado por:

$$\hat{E}(w) = \frac{1}{n}\sum_{k=1}^{n} \|\hat{y}^{(k)}(x^{(k)};w) - y^{(k)}\|^2$$

Aprendizagem supervisionada com descida de gradiente

- Discrepância entre o desejado e o calculado:

$$\delta = \hat{y} - y$$

Para o k-ésimo exemplo: $\delta^{(k)} = \hat{y}\left(x^{(k)}; w\right) - y^{(k)}$

- Problema Linear: $\delta^{(k)} = \underline{w} \,.\, \underline{x}^{(k)} - y^{(k)}$
- Erro individual para o k-ésimo padrão:

$$\|\delta^{(k)}\|^2 = \left\| \begin{pmatrix} \delta_1^k \\ \dots \\ \delta_c^k \end{pmatrix} \right\|^2$$

Aprendizagem supervisionada com descida de gradiente

- Erro global aproximado:

$$\hat{E}(w) = \frac{1}{n}\sum_{k=1}^{n} \|\delta^{(k)}\|^2 = \frac{1}{n}\sum_{i=1}^{c}\sum_{k=1}^{n} [\delta_i^{(k)}]^2$$

$$= \frac{1}{n}\sum_{k=1}^{n} \{[\delta_1^{(1)2} + \ldots + \delta_c^{(1)2}] + \ldots + [\delta_1^{(n)2} + \ldots + \delta_c^{(n)2}]\}$$

$$= \frac{1}{n}\sum_{k=1}^{n} \{[\delta_1^{(1)2} + \ldots + \delta_1^{(n)2}] + \ldots + [\delta_c^{(1)2} + \ldots + \delta_c^{(n)2}]\}$$

Utilização da Função de Erro

- Aprendizagem Estocástica:
  - Após a apresentação de cada padrão individual $x^{(k)}$ (em ordem aleatória) ⇒ $E^{(k)}(w)$ e ajustar *w* usando o princípio de descida de gradiente.
- Aprendizagem em Lote (batch):
  - Utilizar o erro acumulado sobre todos os *n* padrões:

$$E(w)=\frac{1}{n}\sum_{k=1}^{n} E^{(k)}(w)$$

$$w(\tau+1)=w(\tau)+\eta(-\nabla E(w))$$

$$E^{(k)}(W_i)=\sum_{i=1}^{c}[\delta^{(k)}]^2=\sum_{i=1}^{c}[w_i x^{(k)}-y_i^{(k)}]^2$$

$$gradiente\ \nabla E^{(k)}(w_i)=\begin{pmatrix}\partial E^{(k)}(w_i)/\partial w_{i0}\\ \partial E^{(k)}(w_i)/\partial w_{ij}\\ \dots\\ \partial E^{(k)}(w_i)/\partial w_{id}\end{pmatrix}$$

$$\partial E^{(k)}(w_i)/\partial w_{ij}=\frac{\partial\sum_{i=1}^{c}(\delta^{(k)})^2}{\partial w_{ij}}=\frac{\sum_{i=1}^{c}\partial(\delta^{(k)})^2}{\partial w_{ij}}=\sum_{i=1}^{c}\partial\frac{[w_i x^{(k)}-y_i]^2}{\partial w_{ij}}=$$

$$=2.\sum_{i=1}^{c}\partial[w_i x^{(k)}-y_i].\partial\frac{[w_{i0}x_0^{(k)}+\dots+w_{ij}x_j^{(k)}+\dots+w_{id}x_d^{(k)}]}{\partial w_{ij}}$$

$$=2.\sum_{i=1}^{c}\delta_i^{(k)}.x_j\ \Rightarrow\ \nabla E_i^{(k)}=2.\delta^{(k)}.x_j$$

$$w_{ij}(\tau+1)=w_{ij}(\tau)+\eta'[-2\delta_i^{(k)}.x_j]$$
$$=w_{ij}(\tau)-\eta[\delta_i^{(k)}.x_j]$$

Então:

$$E^{(k)}(W_i) = \sum_{i=1}^{c} [w_i x^{(k)} - y_i^{(k)}]^2$$

$$\nabla E^{(k)}(w_i) = \begin{pmatrix} \partial E^{(k)}(w_i)/\partial w_{i0} \\ \partial E^{(k)}(w_i)/\partial w_{ij} \\ \dots \\ \partial E^{(k)}(w_i)/\partial w_{id} \end{pmatrix} = 2.\delta^{(k)}.x_j$$

$$w_{ij}(\tau+1) = w_{ij}(\tau) + \eta'[-2\delta_i^{(k)}.x_j]$$
$$= w_{ij}(\tau) - \eta[\delta_i^{(k)}.x_j]$$

Treino

- Estocástico:

Para $k=1,\ldots,n$ (aleatório):
  Para $i=1,\ldots,c$:
    Para $j=0,\ldots,d$:
      $$w(\tau+1)=wij(\tau)-\eta\,\delta_i^{(k)}\,x_j^{(k)}$$

- Em Lote (Batch):

$$E=\sum_{k=1}^{n}E^{(k)} \qquad \partial E(w_i)=2.\sum_{k=1}^{n}\delta_i^{(k)}\,X_j^{(k)}$$

$$W_{ij}(\tau+1)=W_{ij}(\tau)-\eta.\frac{1}{n}\sum_{k=1}^{n}\delta_i^{(k)}\,X_j^{(k)}$$

Momento

- O Momento é uma heurística para melhorar a convergência.

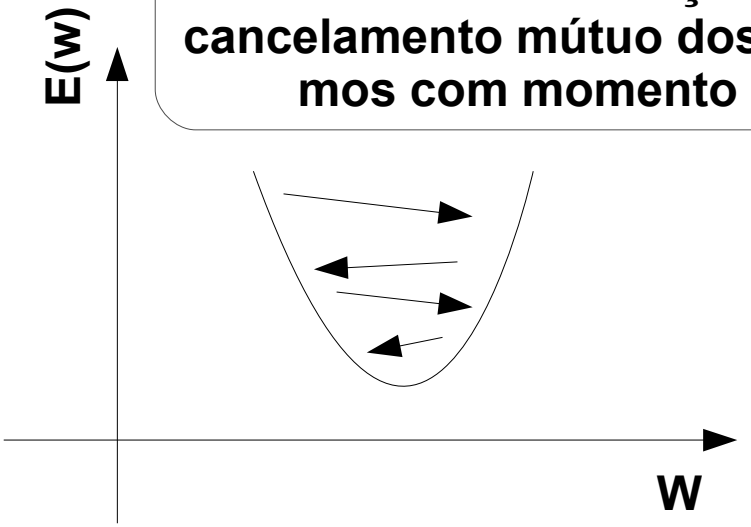
$$\underline{W}(\tau+1)=\underline{W}(\tau)+\Delta\underline{W}(\tau)$$
$$\Delta\underline{W}(\tau)=-\eta\nabla E(\underline{W}(\tau))+\mu\Delta\underline{W}(\tau+1)$$

Parâmetro de Momento > 0

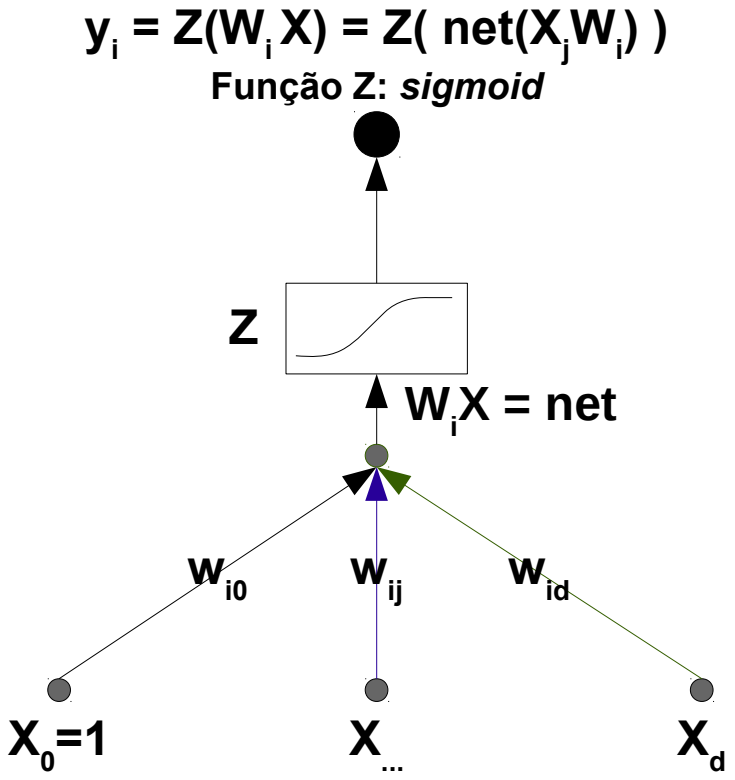
Efeito sem momento: μ = 0
Desaceleração com η fixo

Efeito com momento: μ > 0
Menor desaceleração

Com e sem momento.
Em caso de oscilação, cancelamento mútuo dos termos com momento



Descida de Gradiente



- Regra de Delta:

$$W_{ij}(\tau+1) = W_{ij}(\tau) - \eta \nabla E(\underline{w})\Big|_{\underline{w}(\tau)}$$

$$E_{(W_i)}^{(k)} = \delta_i^{(k)} = \hat{y}_i^{(k)} - y_i^{(k)} = Z(W_i X^{(k)}) - y_i^{(k)}$$

$$\nabla E(W_{ij}) = \frac{\partial E}{\partial W_{ij}} = 2.\delta_i^{(k)}(W_i).Z'(net).X_j^{(k)}$$

$$\frac{\partial E}{\partial W_{ij}} = 2.\delta_i^{(k)}(W_i).(1-net').X_j^{(k)}$$