Machine Learning (CSE 446): Learning as Minimizing Loss (continued)

Noah Smith

© 2017

University of Washington
nasmith@cs.washington.edu

October 25, 2017

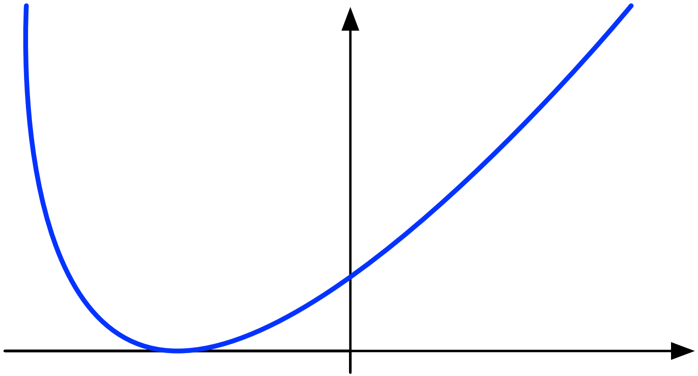
warning

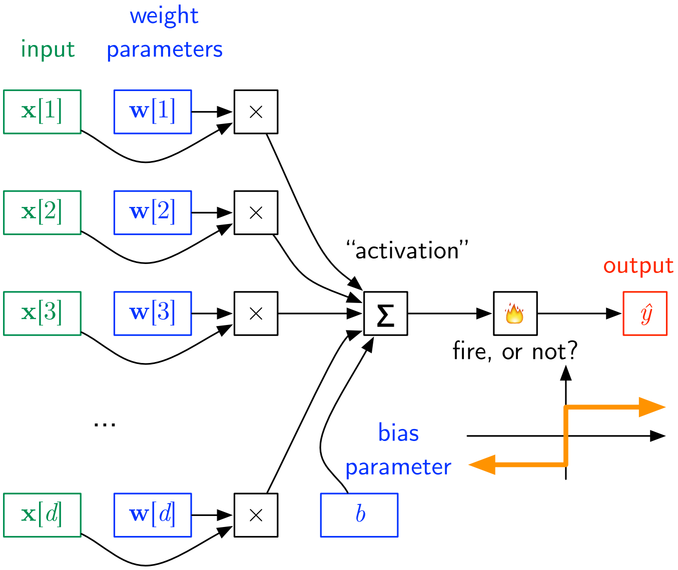
Gradient Descent

Data: function $F : \mathbb{R}^d \to \mathbb{R}$, number of iterations K, step sizes $\langle \eta^{(1)}, \ldots, \eta^{(K)} \rangle$
Result: $\mathbf{z} \in \mathbb{R}^d$
initialize: $\mathbf{z}^{(0)} = \mathbf{0}$;
for $k \in \{1, \ldots, K\}$ **do**
    $\mathbf{g}^{(k)} = \nabla_{\mathbf{z}} F(\mathbf{z}^{(k-1)})$;
    $\mathbf{z}^{(k)} = \mathbf{z}^{(k-1)} - \eta^{(k)} \cdot \mathbf{g}^{(k)}$;
end
return $\mathbf{z}^{(K)}$;

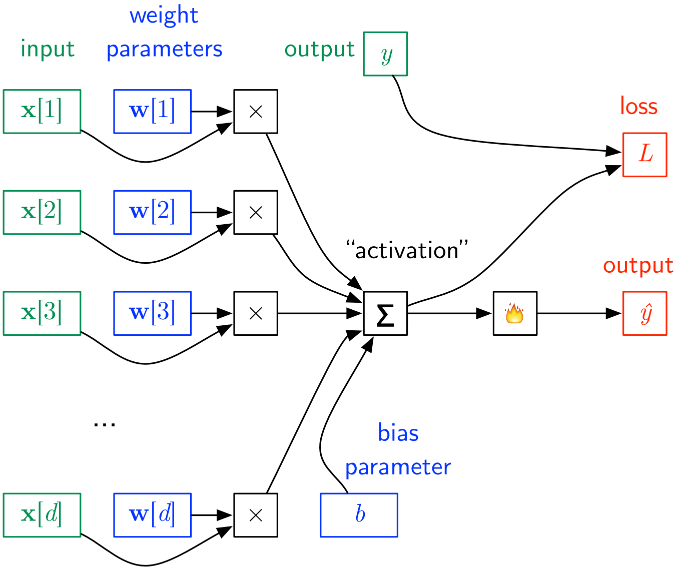
Algorithm 1: GradientDescent

Gradient Descent

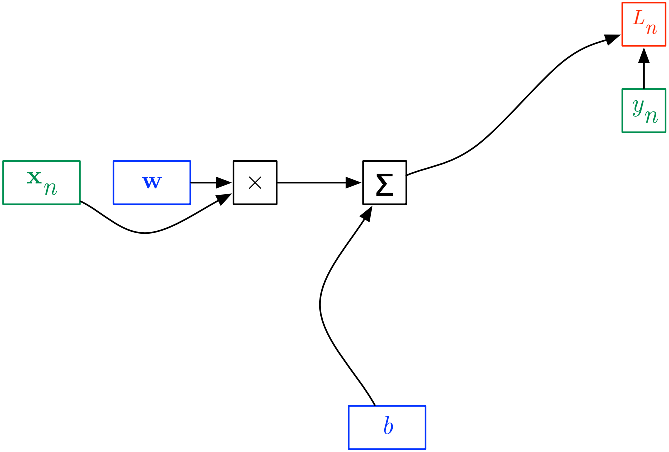




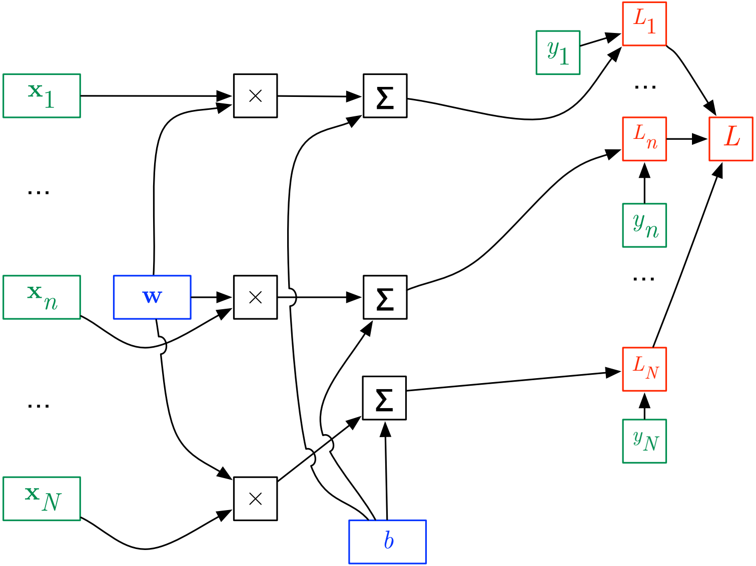
Classification



Training (one example)



Training (one example)

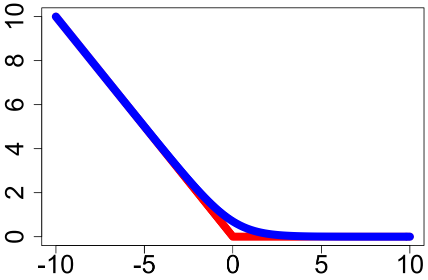


$\mathbf{x}_1$
$\mathbf{x}_n$
$\mathbf{x}_N$
$\mathbf{w}$
b
$\times$
$\times$
$\times$
Σ
Σ
Σ
y_1
y_n
y_N
L_1
L_n
L_N
L

Training (D)

Log Loss

The log loss is continuous, convex, and differentiable. It is closely related to the perceptron loss.



$$L_n(\mathbf{w}) = \log\left(1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right)$$

First Derivative of the Log Loss for One Example

$$\frac{\partial L_n}{\partial w[j]} = \frac{\partial}{\partial w[j]} \log\left(1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right)$$

First Derivative of the Log Loss for One Example

$$\begin{aligned}\frac{\partial L_n}{\partial w[j]} &= \frac{\partial}{\partial w[j]} \log\left(1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right) \\ &= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right]\end{aligned}$$

Remember: $\frac{\partial}{\partial x} \log f(x) = \frac{\frac{\partial}{\partial x} f(x)}{f(x)}$.

First Derivative of the Log Loss for One Example

$$
\begin{aligned}
\frac{\partial L_n}{\partial w[j]} &= \frac{\partial}{\partial w[j]} \log\left(1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right) \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right] \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)
\end{aligned}
$$

First Derivative of the Log Loss for One Example

$$
\begin{aligned}
\frac{\partial L_n}{\partial w[j]} &= \frac{\partial}{\partial w[j]} \log\left(1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right) \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right] \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right) \\
&= \frac{\exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right]
\end{aligned}
$$

Remember: $\frac{\partial}{\partial x} \exp f(x) = \exp f(x) \cdot \frac{\partial}{\partial x} f(x)$.

First Derivative of the Log Loss for One Example

$$
\begin{aligned}
\frac{\partial L_n}{\partial w[j]} &= \frac{\partial}{\partial w[j]} \log\left(1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right) \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right] \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right) \\
&= \frac{\exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right] \\
&= \frac{\exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n[j]
\end{aligned}
$$

First Derivative of the Log Loss for One Example

$$
\begin{aligned}
\frac{\partial L_n}{\partial w[j]} &= \frac{\partial}{\partial w[j]} \log\left(1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right) \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)\right] \\
&= \frac{1}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right) \\
&= \frac{\exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot \frac{\partial}{\partial w[j]} \left[-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right] \\
&= \frac{\exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)}{1 + \exp\left(-y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n[j] \\
&= \frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n[j]
\end{aligned}
$$

$\frac{\exp x}{1+\exp x} = \frac{\exp x}{1+\exp x} \cdot \frac{\exp -x}{\exp -x} = \frac{1}{1+\exp -x}$.

Gradient of the Log Loss, All Examples

One example:

$$\frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n[j]$$

Sum over all examples:

$$\sum_{n=1}^{N} \frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n[j]$$

Gradient stacks all first derivatives into a vector:

$$\sum_{n=1}^{N} \frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n$$

Perceptron Learning vs. Gradient Descent on Log Loss

Updating rule for the perceptron (single example n):

$$\mathbf{w} \leftarrow \mathbf{w} + [\![\hat{y}_n \neq y_n]\!] \cdot y_n \cdot \mathbf{x}_n$$

Perceptron Learning vs. Gradient Descent on Log Loss

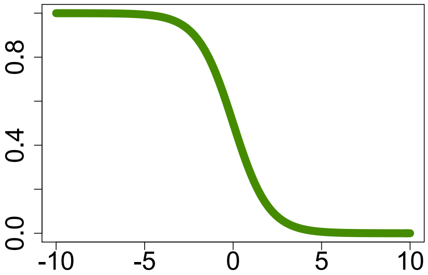
Updating rule for the perceptron (single example n):

$$\mathbf{w} \leftarrow \mathbf{w} + [\![\hat{y}_n \neq y_n]\!] \cdot y_n \cdot \mathbf{x}_n$$

Gradient descent on log loss:

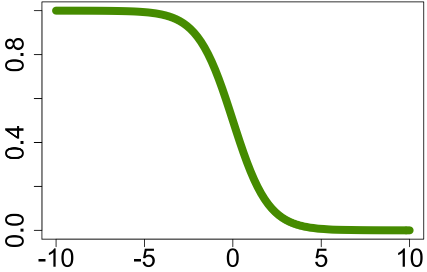
$$\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} - \eta \cdot \sum_{n=1}^{N} \frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n \\
&= \mathbf{w} + \eta \cdot \sum_{n=1}^{N} \frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot y_n \cdot \mathbf{x}_n
\end{aligned}$$

An Interesting Function



$$\sigma(x) = \frac{1}{1 + \exp x}$$

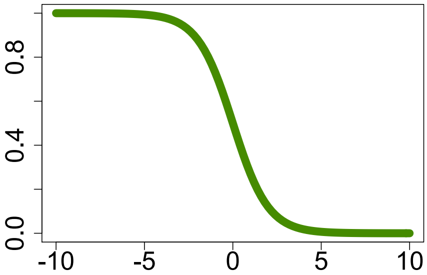
An Interesting Function



$$\sigma(x) = \frac{1}{1 + \exp x}$$

What we plug in to this function is $y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)$.

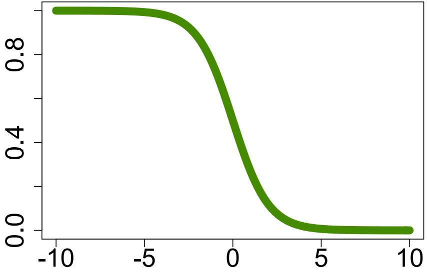
An Interesting Function



$$\sigma(x) = \frac{1}{1 + \exp x}$$

What we plug in to this function is $y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)$.
If example n is getting classified correctly, $\sigma(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b))$ goes to zero.

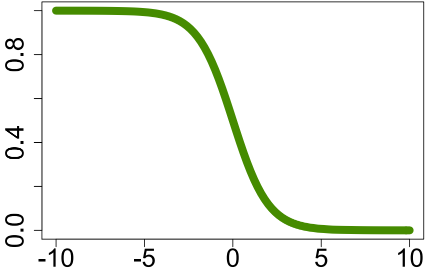
An Interesting Function



$$\sigma(x) = \frac{1}{1 + \exp x}$$

What we plug in to this function is $y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)$.
If example n is getting classified correctly, $\sigma(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b))$ goes to zero.
If example n is getting classified incorrectly, $\sigma(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b))$ goes to one.

An Interesting Function



$$\sigma(x) = \frac{1}{1 + \exp x}$$

What we plug in to this function is $y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)$.
If example n is getting classified correctly, $\sigma(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b))$ goes to zero.
If example n is getting classified incorrectly, $\sigma(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b))$ goes to one.
Not "is it wrong?" but instead "how wrong is it?"

Perceptron Learning vs. Gradient Descent on Log Loss

Updating rule for the perceptron (single example n):

$$\mathbf{w} \leftarrow \mathbf{w} + [\![\hat{y}_n \neq y_n]\!] \cdot y_n \cdot \mathbf{x}_n$$

Gradient descent on log loss:

$$\begin{aligned}
\mathbf{w} \leftarrow \mathbf{w} - \eta \cdot \sum_{n=1}^{N} \frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot -y_n \cdot \mathbf{x}_n \\
= \mathbf{w} + \eta \cdot \sum_{n=1}^{N} \frac{1}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)} \cdot y_n \cdot \mathbf{x}_n \\
= \mathbf{w} + \sum_{n=1}^{N} \underbrace{\frac{\eta}{1 + \exp\left(y_n \cdot (\mathbf{w} \cdot \mathbf{x}_n + b)\right)}}_{\text{“soft” error test}} \cdot y_n \cdot \mathbf{x}_n
\end{aligned}$$

Logistic Regression

Classifier:

$$f(\mathbf{x}) = \mathsf{sign}\left(\mathbf{w} \cdot \mathbf{x} + b\right)$$

(Identical to the neuron-inspired classifier we trained using the perceptron!)

To learn, solve:

$$\min_{\mathbf{w},b} \frac{1}{N} \sum_{n=1}^{N} \underbrace{\log\left(1 + \exp\left(-y_n \cdot \left(\mathbf{w} \cdot \mathbf{x}_n + b\right)\right)\right)}_{\text{log loss}} + R(\mathbf{w}, b)$$

Linear Regression

For *continuous* output y.

Predictor:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

To learn, solve:

$$\min_{\mathbf{w}, b} \frac{1}{N} \sum_{n=1}^{N} \underbrace{(y_n - (\mathbf{w} \cdot \mathbf{x}_n + b))^2}_{\text{squared loss}} + R(\mathbf{w}, b)$$

Coming Soon

- Gradient descent is usually too slow; we can do better.
- Is there a deeper connection between perceptron learning and gradient descent?
- More loss functions!
- More activation functions!
- More regularization functions!
- A probabilistic interpretation of logistic regression and linear regression.