Introduction to Machine Learning
NPFL 054

http://ufal.mff.cuni.cz/course/npfl054

Barbora Hladká
hladka@ufal.mff.cuni.cz

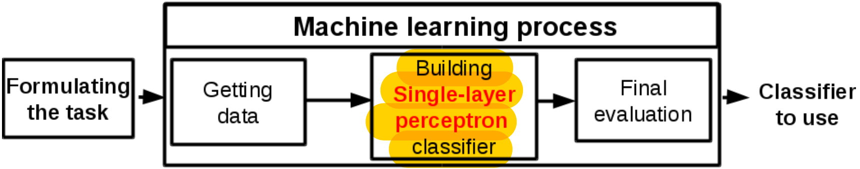
Martin Holub
holub@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Outline

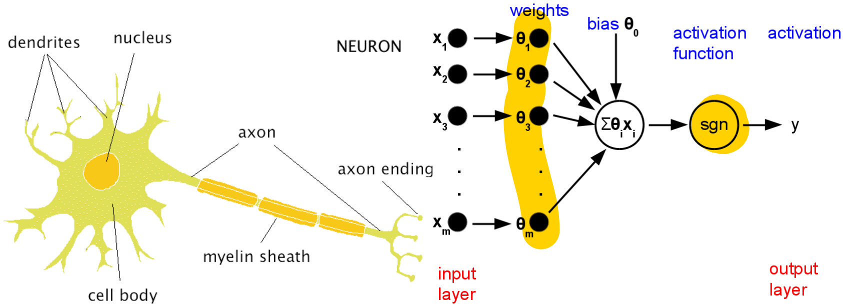
- **Single-layer perceptron**

Single-layer perceptron



Single-layer perceptron (SLP)

biological inspiration	machine learning
neuron	SLP algorithm
other neurons	$\mathbf{x} = \langle x_1, \ldots, x_m \rangle$
connection weights	feature weights $\Theta_1, \ldots, \Theta_m$
amount of neuron activation	$\sum_{i=1}^{m} x_i \Theta_i$
"firing" neuron	output positive classification
"not firing" neuron	output negative classification



- is **on-line learning**
  - Look at one example, process it and go to the next example
- is **error-driven learning**
  - One example at a time
  - Make prediction, compare it with true prediction
  - Update Θ if different

Binary classification with SLP

Neuron

- **Input**: instance $\mathbf{x}$
- **Incoming connections**: feature weights $\Theta_1, \ldots, \Theta_m$, $\boldsymbol{\Theta} = < \Theta_1, \ldots, \Theta_m >$
- **Action**: compute activation $a = \boldsymbol{\Theta}^{\mathrm{T}}\mathbf{x}$
- **Output**: if $a > 0$ output $+1$ otherwise -1

Prefer **non-zero threshold** $\boldsymbol{\Theta}^{\mathrm{T}}\mathbf{x} > \Delta$

- Introduce a **bias** term Θ_0 into the neuron.
- Then $a = \Theta_0 + \boldsymbol{\Theta}^{T}\mathbf{x}$
- **Output**: if $a > 0$ output $+1$ otherwise -1

Binary classification with SLP
Training

$Data = \{\langle \mathbf{x}, y\rangle : \mathbf{x} = \langle x_1, \dots, x_m\rangle, y \in \{-1, +1\}\}$

$\Theta_i \leftarrow 0$ for all $i = 1, \dots, m$ // initialize weights
$\Theta_0 \leftarrow 0$ // initialize bias
    for $iter = 1 \dots$ $\textcolor{red}{MaxIter}$ do
        for all $\langle \mathbf{x}, y \rangle \in$ $\textcolor{red}{Data}$ do
            $a \leftarrow \Theta_0 + \mathbf{\Theta}^T\mathbf{x}$ // compute activation for $\mathbf{x}$
            if $ya \leq 0$ then // update weights and bias
                $\Theta_i \leftarrow \Theta_i + yx_i$ for all $i = 1, \dots, m$
                $\Theta_0 \leftarrow \Theta_0 + y$
            end if
        end for
    end for
return Θ_0*, $\mathbf{\Theta}^\star$

Binary classification with SLP
Test

1. $a \leftarrow \Theta_0^\star + \mathbf{\Theta}^{\star T}\mathbf{x}$ // compute activation for $\mathbf{x}$
2. return sgn(a)

Update Θ

If we can see the given example in the future, we should do a better job.

For illustration:

- Assume positive example $\mathbf{x}$ ($\langle \mathbf{x}, +1 \rangle$) and current Θ_0 and $\mathbf{\Theta}$.
- Do prediction and $y(\Theta_0 + \mathbf{\Theta}^T \mathbf{x}) < 0$, i.e. misclassification
- Update Θ_0 and $\mathbf{\Theta}$
  - $\Theta_0' = \Theta_0 + 1$
  - $\Theta_i' = \Theta_i + 1 * x_i,\ i = 1, \dots, m$
- Process the next example which is by chance the same example $\mathbf{x}$
- Compute $a' = \Theta_0' + (\mathbf{\Theta'})^T \mathbf{x} = \Theta_0 + 1 + (\mathbf{\Theta} + \mathbf{x})^T \mathbf{x} =$
  $\Theta_0 + 1 + \mathbf{\Theta}^T \mathbf{x} + \mathbf{x}^T \mathbf{x} = \Theta_0 + \mathbf{\Theta}^T \mathbf{x} + 1 + \mathbf{x}^T \mathbf{x} > \Theta_0 + \mathbf{\Theta}^T \mathbf{x} = a$
- $\mathbf{x}$ is a positive example so we have moved the activation in the proper direction

Geometric interpretation of SLP

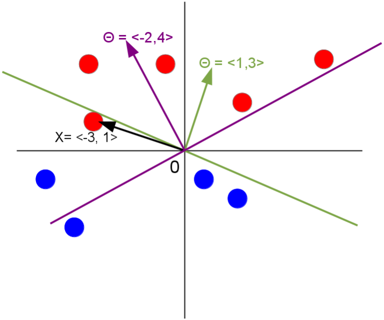
A hyperplane of an m-dimensional space is a flat subset with dimension $m-1$. Any hyperplane can be written as the set of points $\mathbf{x}$ satisfying

$$\Theta_0 + \mathbf{\Theta}^T\mathbf{x} = 0, \text{ where } \mathbf{\Theta} = \begin{pmatrix} \Theta_1 \\ \dots \\ \Theta_m \end{pmatrix}, \mathbf{x} = \langle x_1, \dots, x_m \rangle$$

Geometric interpretation of SLP

Assume $\Theta_0 = 0$

- $\mathbf{\Theta}$ points in the direction of the positive examples and away from the negative examples.



- Having $\mathbf{\Theta}$ normalized, $\mathbf{\Theta}^T\mathbf{x}$ is the length of projection of $\mathbf{x}$ onto $\mathbf{\Theta}$, i.e. the activation of $\mathbf{x}$ with no bias.

Geometric interpretation of SLP

Assume $\Theta_0 \neq 0$

- After the projection is computed, Θ_0 is added to get the overall activation.
- Then $\boldsymbol{\Theta}^T \mathbf{x} + \Theta_0 > 0$?
    - If $\Theta_0 < 0$, Θ_0 shifts the hyperplane away from $\boldsymbol{\Theta}$.
    - If $\Theta_0 > 0$, Θ_0 shifts the hyperplane towards $\boldsymbol{\Theta}$.

Learning parameter `MaxIter`

- many passes $\rightarrow$ overfitting
- only one pass $\rightarrow$ underfitting

Properties of SLP algorithm

- Does the SLP algorithm converge?
  - If the training data IS linearly separable, the SLP algorithm yields a hyperplane that classifies all the training examples correctly.
  - If the training data IS NOT linearly saparable, the SLP algorithm could never possibly classify each example correctly.
- After how many updates the algorithm converges?

Properties of SLP algorithm

Recall the notion of margin of hyperplane

Assume a hyperplane g: $\Theta_0 + \mathbf{\Theta}^{\mathrm{T}}\mathbf{x} = 0$

The **geometric margin** of $\langle \mathbf{x}, y \rangle$ w.r.t. g is

$$\rho_g(\mathbf{x}, y) = y(\Theta_0 + \mathbf{\Theta}^T\mathbf{x})/||\Theta||$$

The **margin** of *Data* w.r.t. g is

$$\rho_g(Data) = \mathrm{argmin}_{\langle \mathbf{x}, y \rangle \in Data}\rho_g(\mathbf{x}, y)$$

$g^\star = \mathrm{argmax}_g \rho_g(Data)$, $g^\star : \Theta_0^\star + \mathbf{\Theta}^{\star\mathrm{T}}\mathbf{x} = 0$

Let $\gamma = \rho_{g^\star}(Data)$

Perceptron Convergence Theorem

Suppose the perceptron algorithm is run on a linearly separable data set *Data* with margin $\gamma > 0$. Assume that $||\mathbf{x}|| \leq 1$ for all examples in *Data*. Then the algorithm will converge after at most $\frac{1}{\gamma^2}$ updates.

Proof: The perceptron algorithm is trying to find $\mathbf{\Theta}$ that points roughly in the same direction as $\mathbf{\Theta}^\star$. We are interested in the angle α between $\mathbf{\Theta}$ and $\mathbf{\Theta}^\star$. Every time the algorithm makes an update, α changes. Thus we approve that α decreases. We will show that

1. $\mathbf{\Theta}^T\mathbf{\Theta}^\star$ increases a lot
2. $||\mathbf{\Theta}||$ does not increase much

Perceptron Convergence Theorem

$\boldsymbol{\Theta}^0$ is the initial weight vector, $\boldsymbol{\Theta}^k$ is the weight vector after k updates.

1. We will show that $\boldsymbol{\Theta}^\star\boldsymbol{\Theta}^k$ grows as a function of k:

$$\boldsymbol{\Theta}^\star\boldsymbol{\Theta}^k \overset{\text{definition of } \boldsymbol{\Theta}^k}{=} \boldsymbol{\Theta}^\star(\boldsymbol{\Theta}^{k-1} + y\mathbf{x}) \overset{\text{vector algebra}}{=}$$
$$= \boldsymbol{\Theta}^\star\boldsymbol{\Theta}^{k-1} + y\boldsymbol{\Theta}^\star\mathbf{x} \overset{\boldsymbol{\Theta}^\star \text{has margin} \gamma}{\geq} \boldsymbol{\Theta}^\star\boldsymbol{\Theta}^{k-1} + \gamma$$

Therefore $\boldsymbol{\Theta}^\star\boldsymbol{\Theta}^k \geq k\gamma$

Perceptron Convergence Theorem

2. We update $\mathbf{\Theta}^k$ because $y(\mathbf{\Theta}^{k-1})^T\mathbf{x} < 0$

$$||\mathbf{\Theta}^k||^2 = ||\mathbf{\Theta}^{k-1} + y\mathbf{x}||^2 \overset{\text{quadratic rule of vectors}}{=}$$
$$||\mathbf{\Theta}^{k-1}||^2 + y^2||\mathbf{x}||^2 + 2y\mathbf{\Theta}^{k-1}\mathbf{x} \overset{\text{assumption on}||\mathbf{x}||\text{ and } a<0}{\leq} ||\mathbf{\Theta}^{k-1}||^2 + 1 + 0$$

Therefore $||\mathbf{\Theta}^k||^2 \leq k$

Perceptron Convergence Theorem

Putting 1. and 2. together, we can write

$$\sqrt{k} \overset{2.}{\geq} ||\mathbf{\Theta}^k|| \overset{\mathbf{\Theta}^\star \text{ is a unit vector}}{\geq} (\mathbf{\Theta}^\star)^T \mathbf{\Theta}^k \overset{1.}{\geq} k\gamma \Rightarrow k \leq 1/\gamma^2$$

Perceptron Convergence Theorem

- The proof says that if the perceptron gets linearly separable data with γ, then it will converge to a solution that separates the data.
- The proof does not speak about the solution, other than the fact that it separates the data. The proof makes use of the maximum margin hyperplane. But the perceptron is not guaranteed to find m.m. hyperplane.

Mutliclass classification with SLP
Training

$Y = \{1, \dots, k\}$. There is a weight vector for each class $\mathbf{\Theta}^1, \dots, \mathbf{\Theta}^k$

1. Initialize weights $\mathbf{\Theta}^k \leftarrow \langle 0, \dots, 0 \rangle$ for all $i = 1, \dots, k$
2. for $iter = 1 \dots MaxIter$ do
3. for all $\langle \mathbf{x}, y \rangle \in Data$ do
4. compute $\mathbf{\Theta}^{i^T} \mathbf{x}$ for all $i = 1, \dots, k$
5. $\hat{y} = argmax_i \; \mathbf{\Theta}^{i^T} \mathbf{x}$
   if y and $\hat{y}$ are different then
6. $\mathbf{\Theta}^y \leftarrow \mathbf{\Theta}^y - \mathbf{x}$
7. $\mathbf{\Theta}^{\hat{y}} \leftarrow \mathbf{\Theta}^{\hat{y}} + \mathbf{x}$
8. end if
9. end for
10. end for
11. return $\mathbf{\Theta}^{1^\star}, \dots, \mathbf{\Theta}^{k^\star}$

Mutliclass classification with SLP
Test

1. `return` $argmax_i \; (\mathbf{\Theta}^{i^\star})^T \mathbf{x}$

References

- Goodfellow, I. – Bengio, Y. – Courville. A. *Deep Learning Book*. 2016. http://www.deeplearningbook.org
- Hal Daume III. A Course on Machine Learning. http://ciml.info