Introduction to Deep Learning: Perceptron Learning

Dripta Mj
Department of Mathematics
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, India

Machine Translation

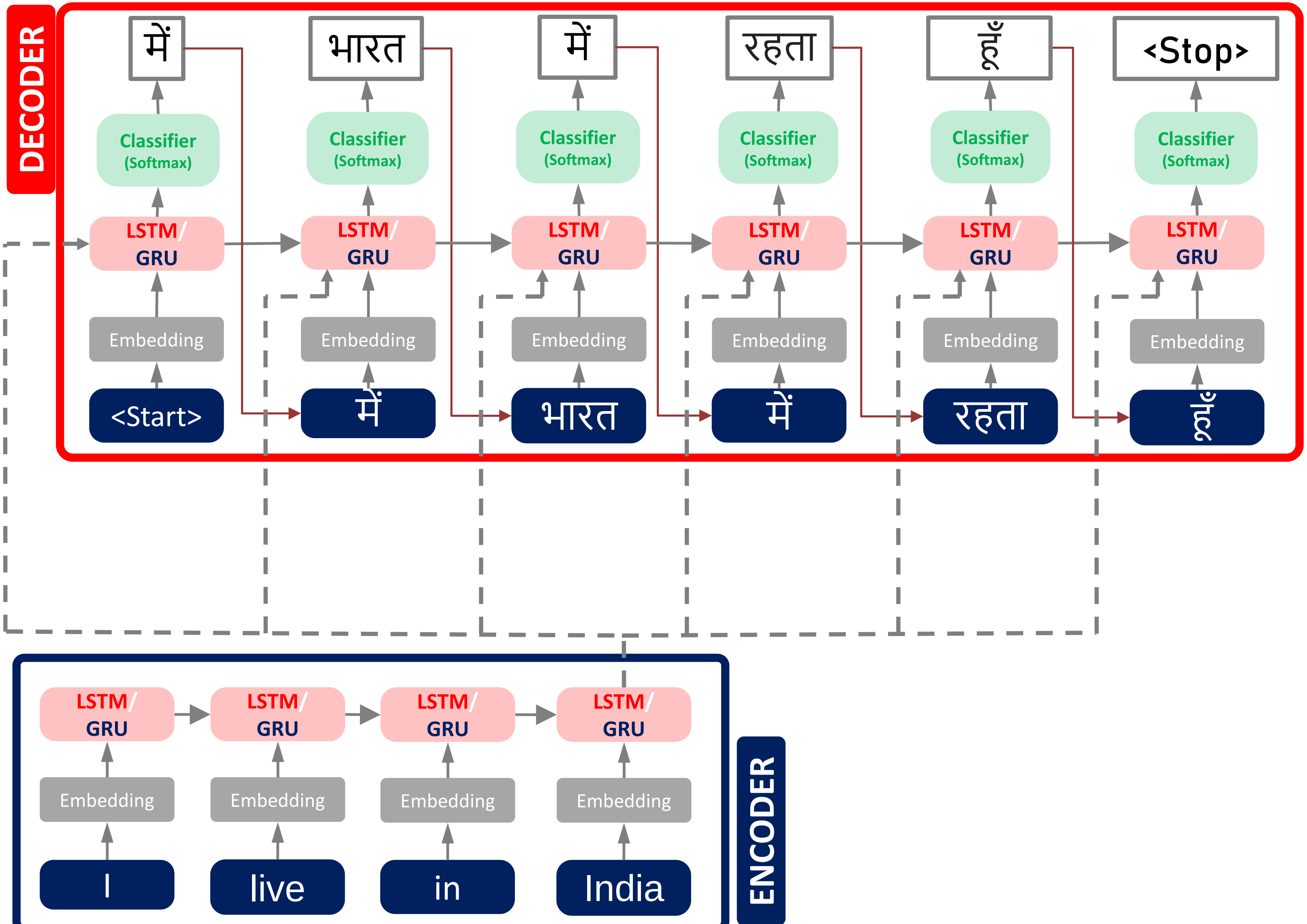
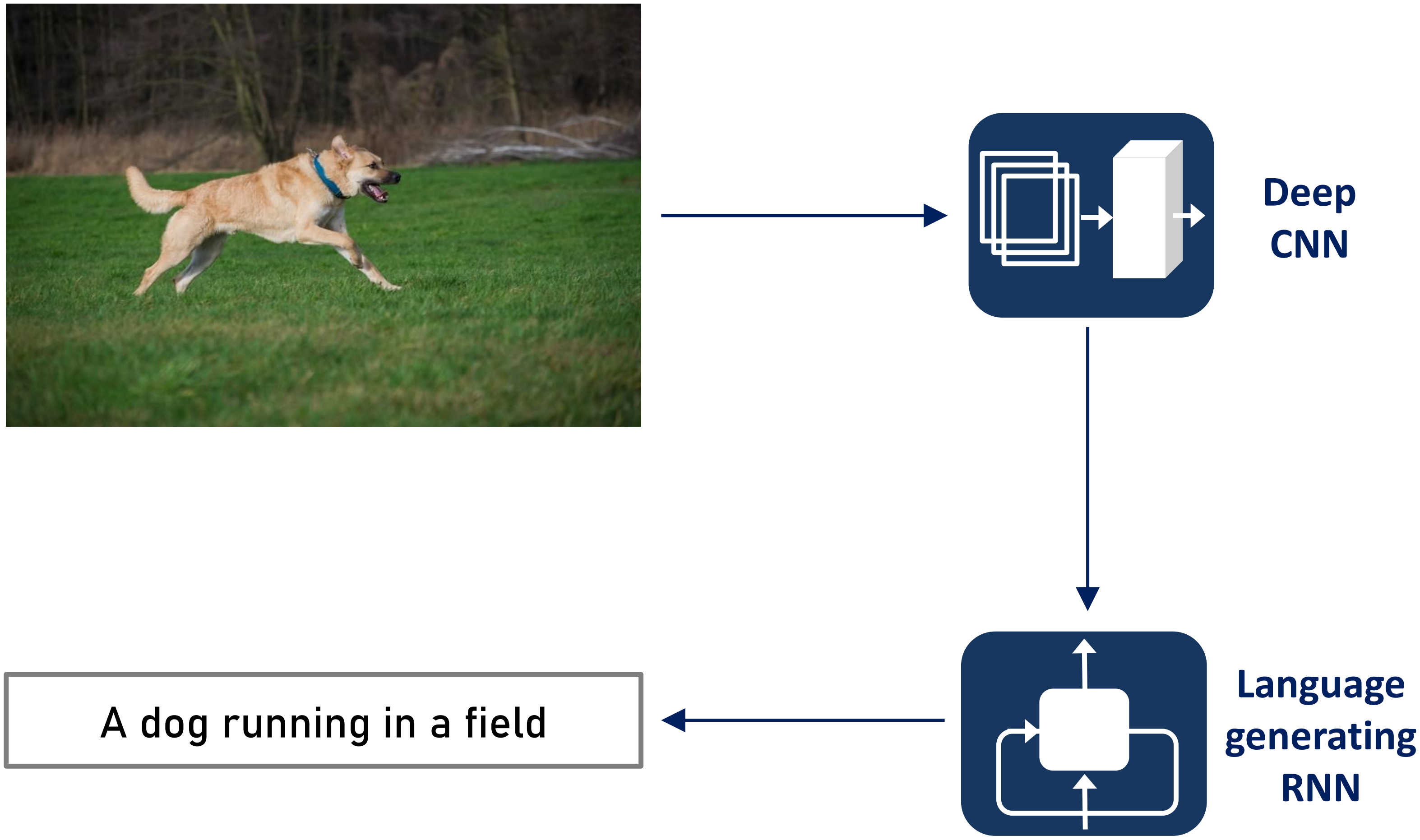
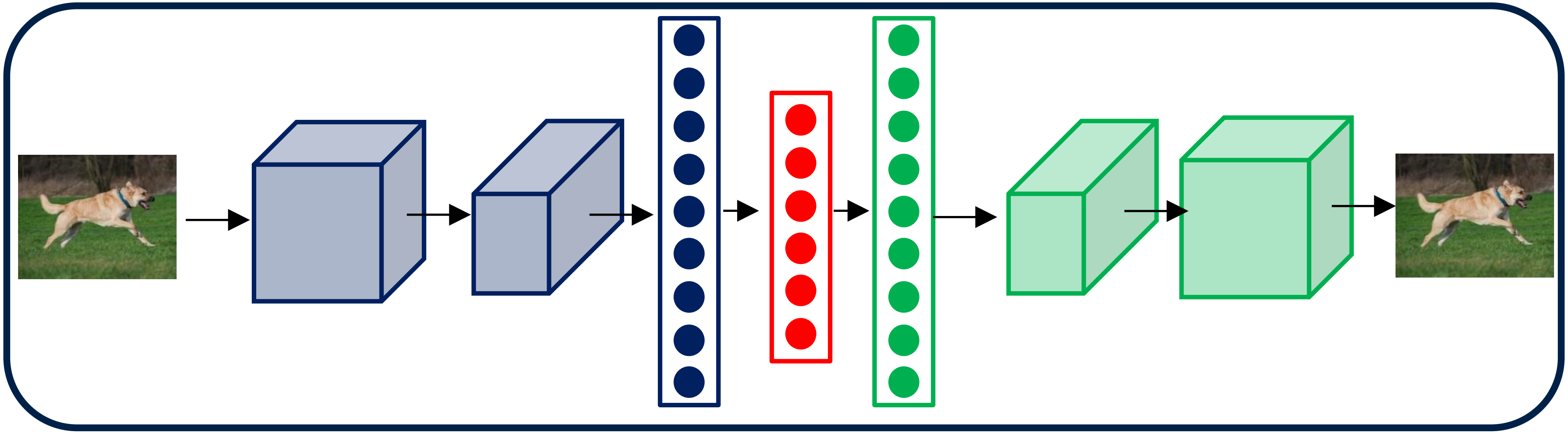


Image captioning



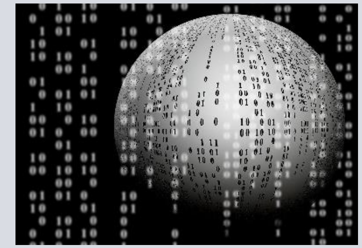
Convolutional Autoencoder



The rise of Deep Learning ....

BIG DATA

- World is data rich!
- Deep learning needs big datasets.



HARDWARE

- Graphics Processing Units

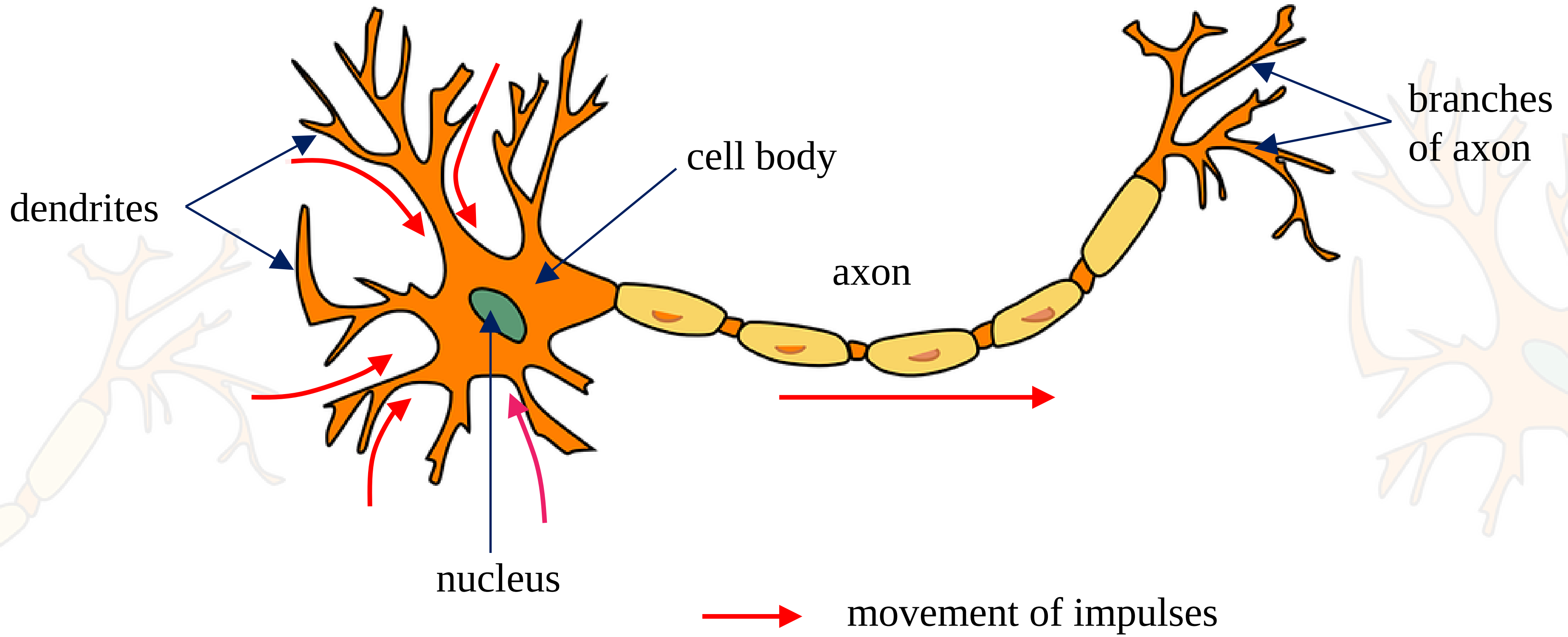
SOFTWARE

- Open source toolboxes
- Efficient implementations

TensorFlow PyTorch

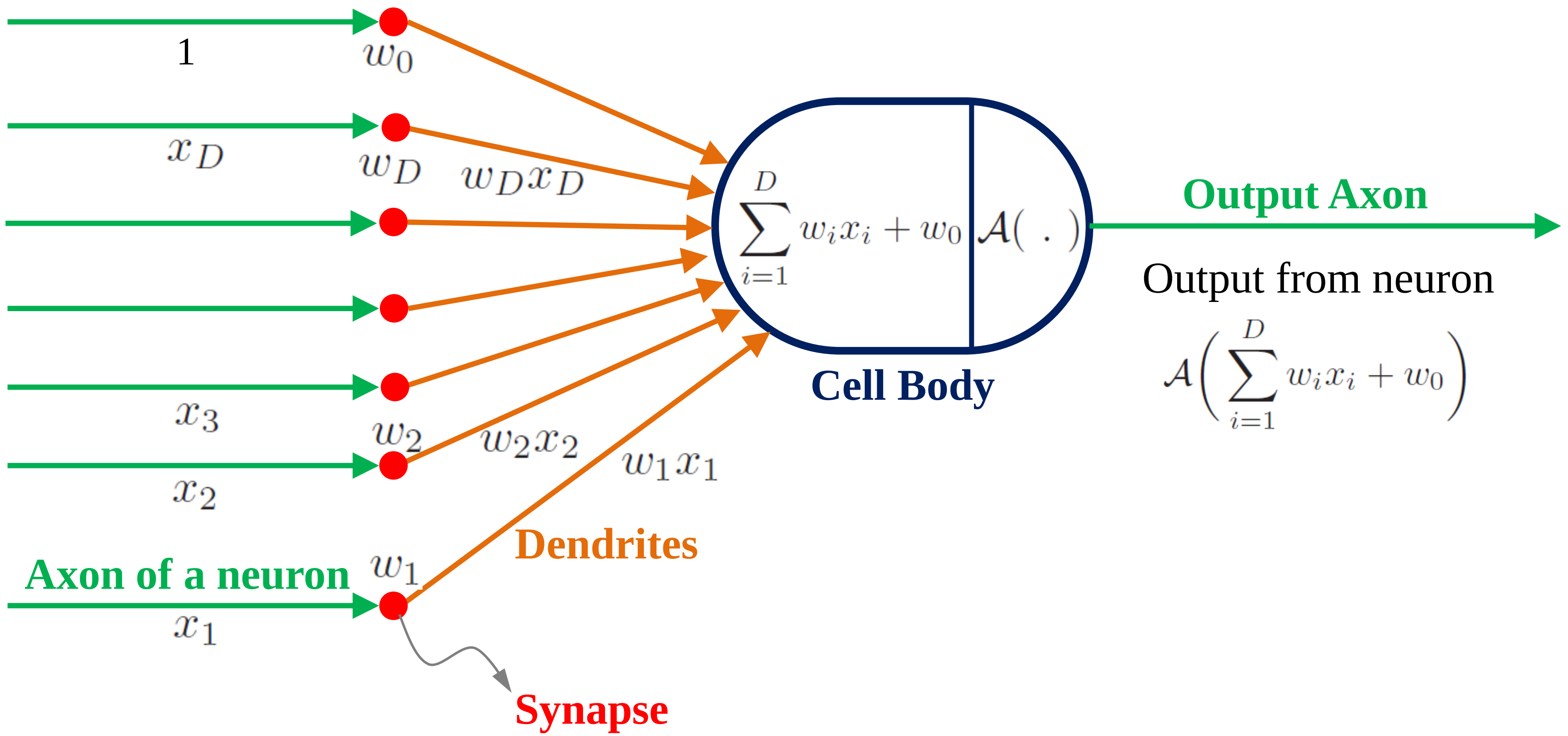
Neuron

- The brain is composed of densely interconnected network of neurons.
- Each neuron has a body, axon, synapses and dendrites.
- A neuron fires if the sum of the weighted signals is greater than a threshold.
- Signals propagate along neurons via axons, synapses and dendrites.

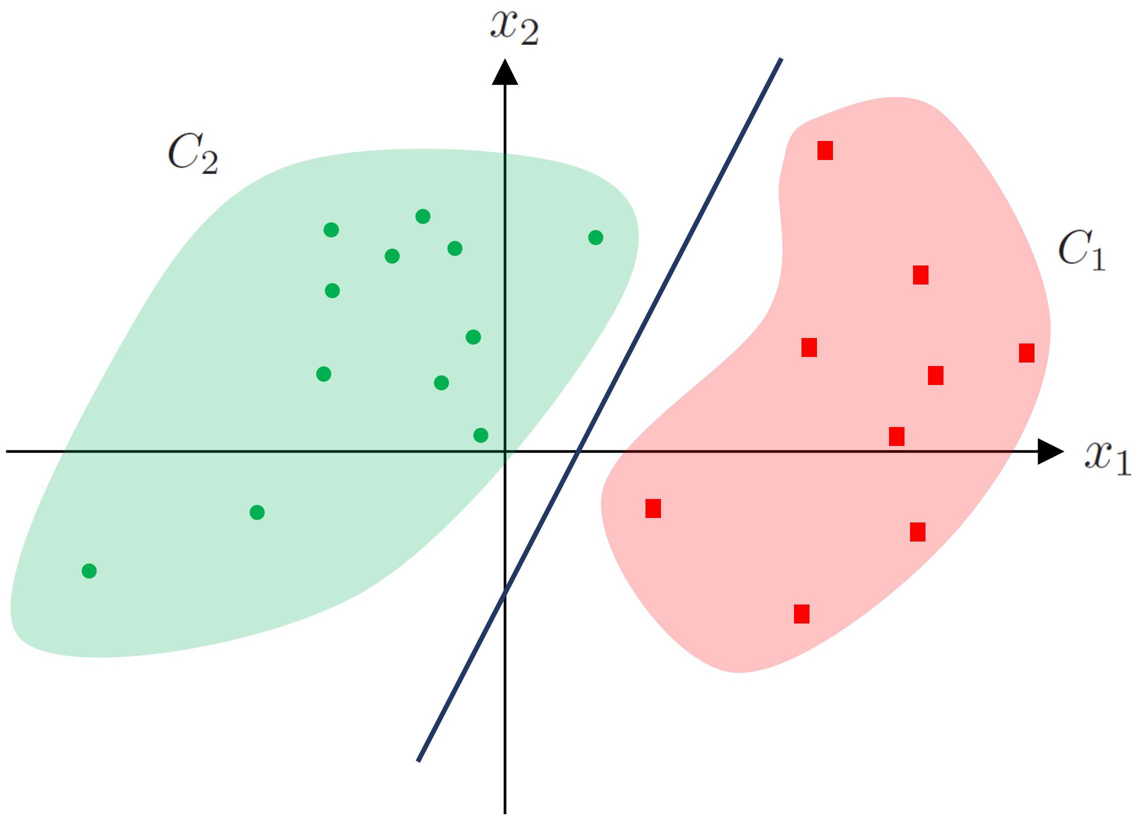


Mathematical model of a neuron

- The threshold activation function "fires" if the weighted sum of the inputs and bias exceeds a certain threshold.

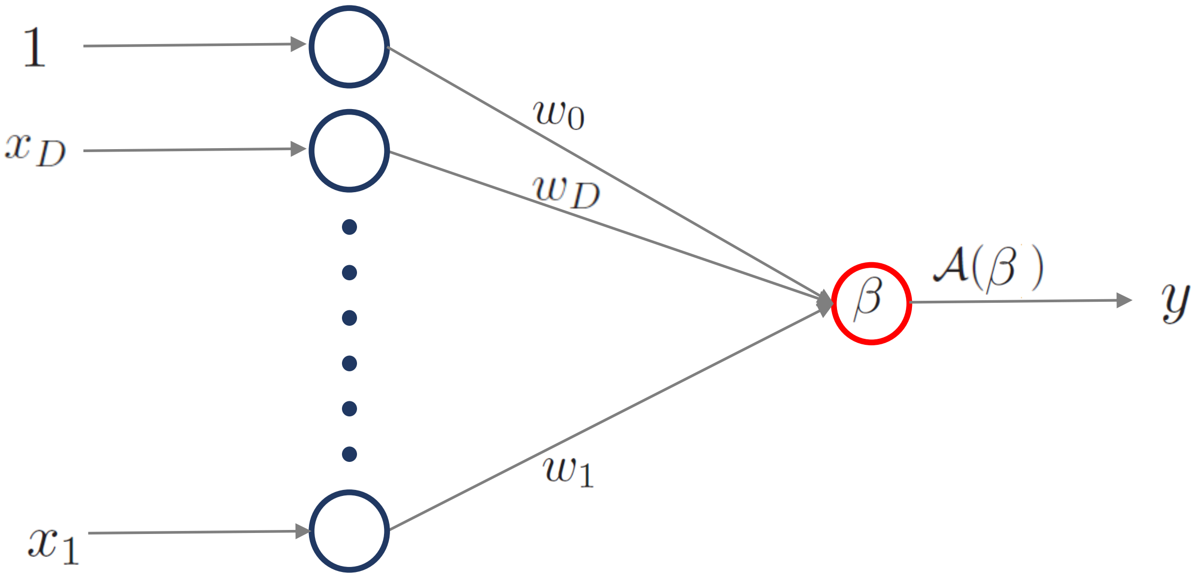


Perceptron Learning



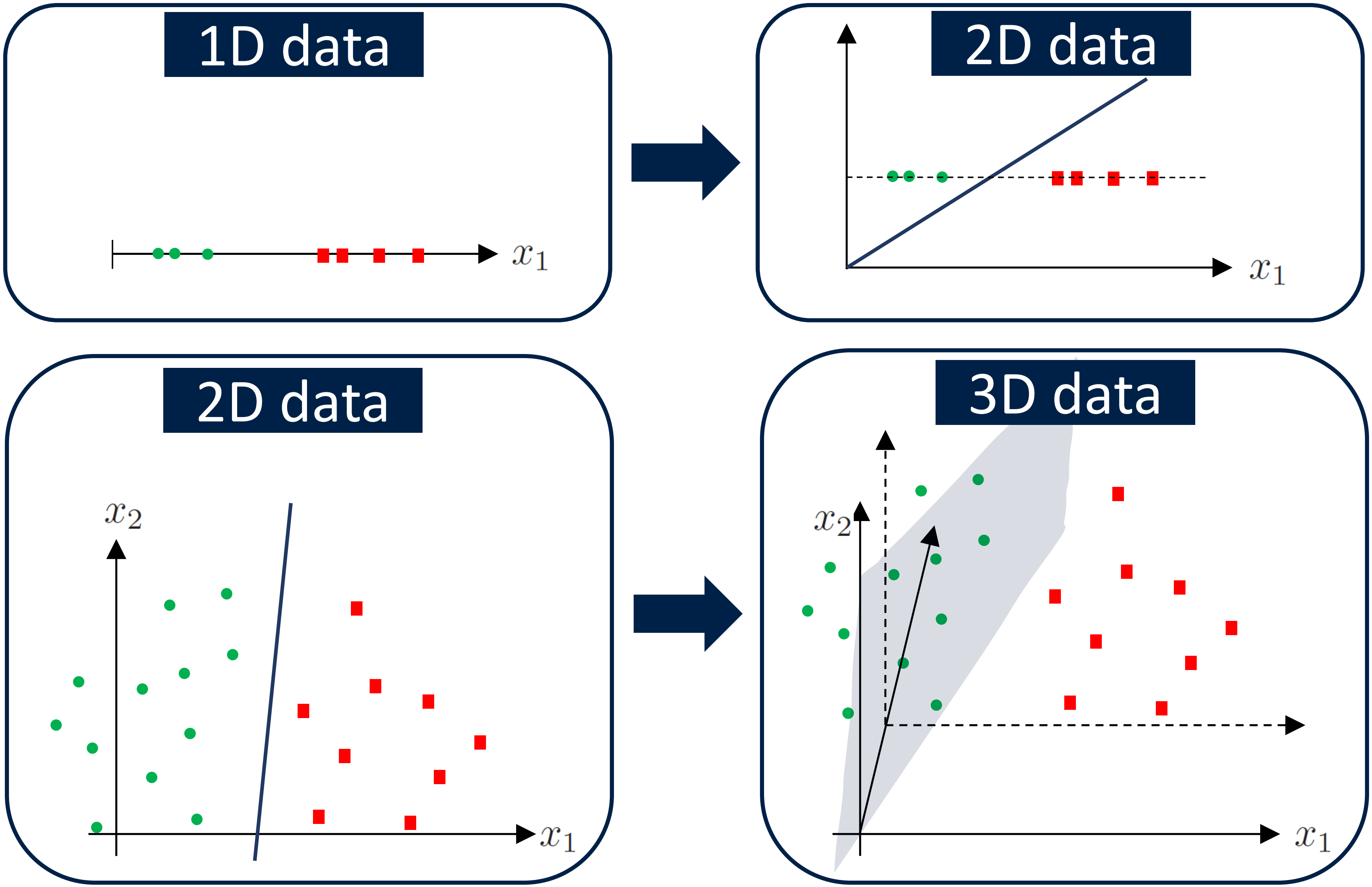
1D Perceptron

- Single layer feed-forward network – only 1 layer of weights is used.



$$\beta = \mathbf{w}^T \mathbf{x}$$

Augmenting input space



Functional margin

- Correct classification:
  - If $\mathbf{x} \in y = 1$, then $\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0$
  - If $\mathbf{x} \in y = -1$, then $\mathbf{w}^{\mathrm{T}}\mathbf{x} < 0$

- The two equations can be jointly written as: $y\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0$

- The functional margin $\widehat{\gamma}_n$ of an example $(\mathbf{x}^{(n)}, y^{(n)})$ with respect to the hyperplane $\mathbf{w}$ is

$$\widehat{\gamma}_n = y^{(n)}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)}\right)$$

- $+$ve $\widehat{\gamma}_n$ means the example is correctly classified.
- $-$ve $\widehat{\gamma}_n$ means the example is incorrectly classified.

Geometric margin

- Geometric margin γ_n of an example is the signed perpendicular distance of the point to the hyperplane

$$\gamma_n = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)}}{||\mathbf{w}||}$$

- Margin is defined as the minimum of the geometric margin

$$\gamma = \min_{\mathcal{D}} |\gamma_n|$$

Approach

- Perceptron algorithm checks if all the training examples are correctly classified with respect to a hyperplane.
  - If an example is misclassified, then the hyperplane is updated.

- If after k updates, a misclassified example $(\mathbf{x}^{(n)}, y^{(n)})$ is encountered, then $\mathbf{w}^{(k)}$ is updated as

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}$$

- What happens if an example is misclassified?
  - If $(\mathbf{x}^{(n)}, y^{(n)})$ is misclassified (after k updates to weights), then

$$\widehat{\gamma}_n = y^{(n)}(\mathbf{w}^{(k)})^{\mathrm{T}}\mathbf{x}^{(n)} < 0.$$

  - Updated weights: $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}$

Approach

– New functional margin $\widehat{\gamma}_n^{(\text{new})}$ is then

$$
\begin{aligned}
\widehat{\gamma}_n^{(\text{new})} &= y^{(n)}\left(\mathbf{w}^{(k+1)}\right)^{\mathrm{T}}\mathbf{x}^{(n)} \\
&= y^{(n)}\left(\mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}\right)^{\mathrm{T}}\mathbf{x}^{(n)} \\
&= y^{(n)}\left(\mathbf{w}^{(k)}\right)^{\mathrm{T}}\mathbf{x}^{(n)} + \left(y^{(n)}\right)^{2}||\mathbf{x}^{(n)}||^{2} \\
&\geq y^{(n)}\left(\mathbf{w}^{(k)}\right)^{\mathrm{T}}\mathbf{x}^{(n)} \\
&\geq \widehat{\gamma}_n^{(\text{old})}
\end{aligned}
$$

- So the new hyperplane $\mathbf{w}^{(k+1)}$ has a larger functional margin than older one $\Rightarrow$ better classification of $(\mathbf{x}^{(n)}, y^{(n)})$.

Algorithm

Initialize: $\mathbf{w} = 0$

while (not converged) **then**

$k = 0$

for $n = 1, 2, .., N$ **do**

if $y^{(n)}((\mathbf{w}^{(k)})^{\mathrm{T}}\mathbf{x}) \leq 0$ **then**

$\mathbf{w} \longleftarrow \mathbf{w} + y^{(n)}x^{(n)}$

$k \longleftarrow k + 1$

end if

end for

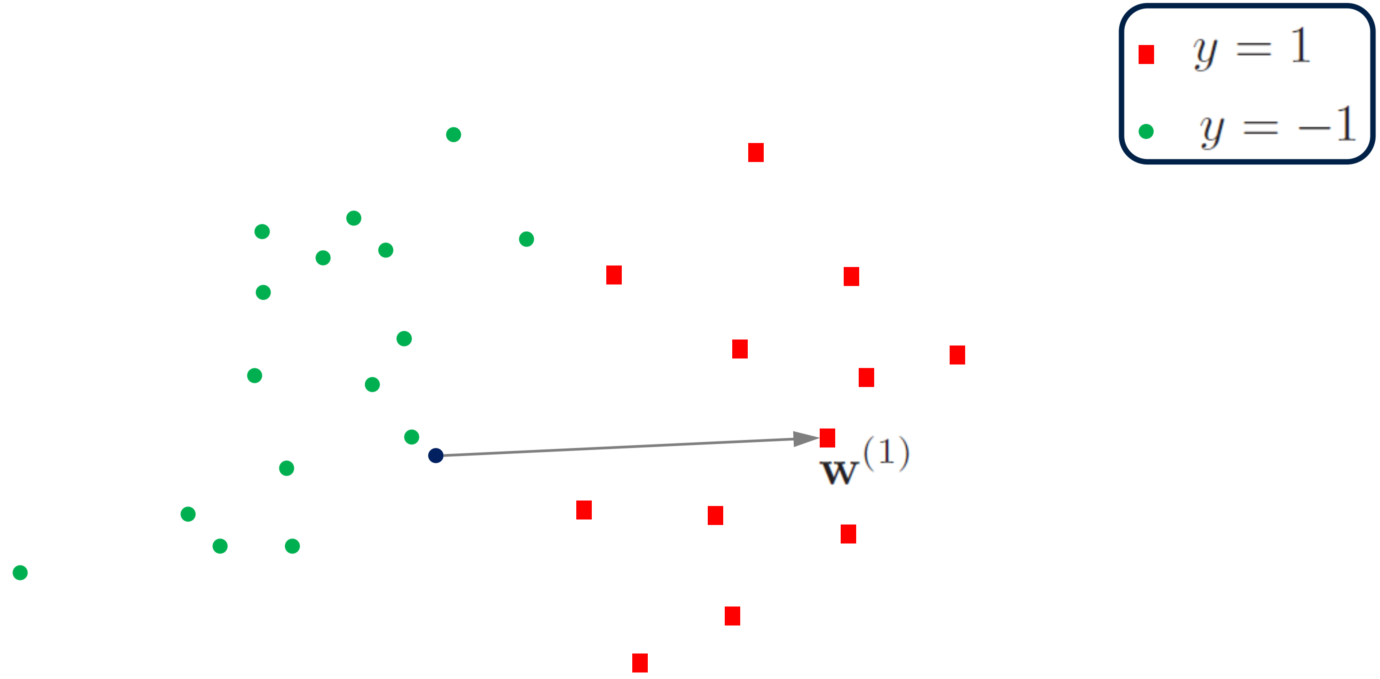
if $k = 0$ **then**

break

end if

end loop

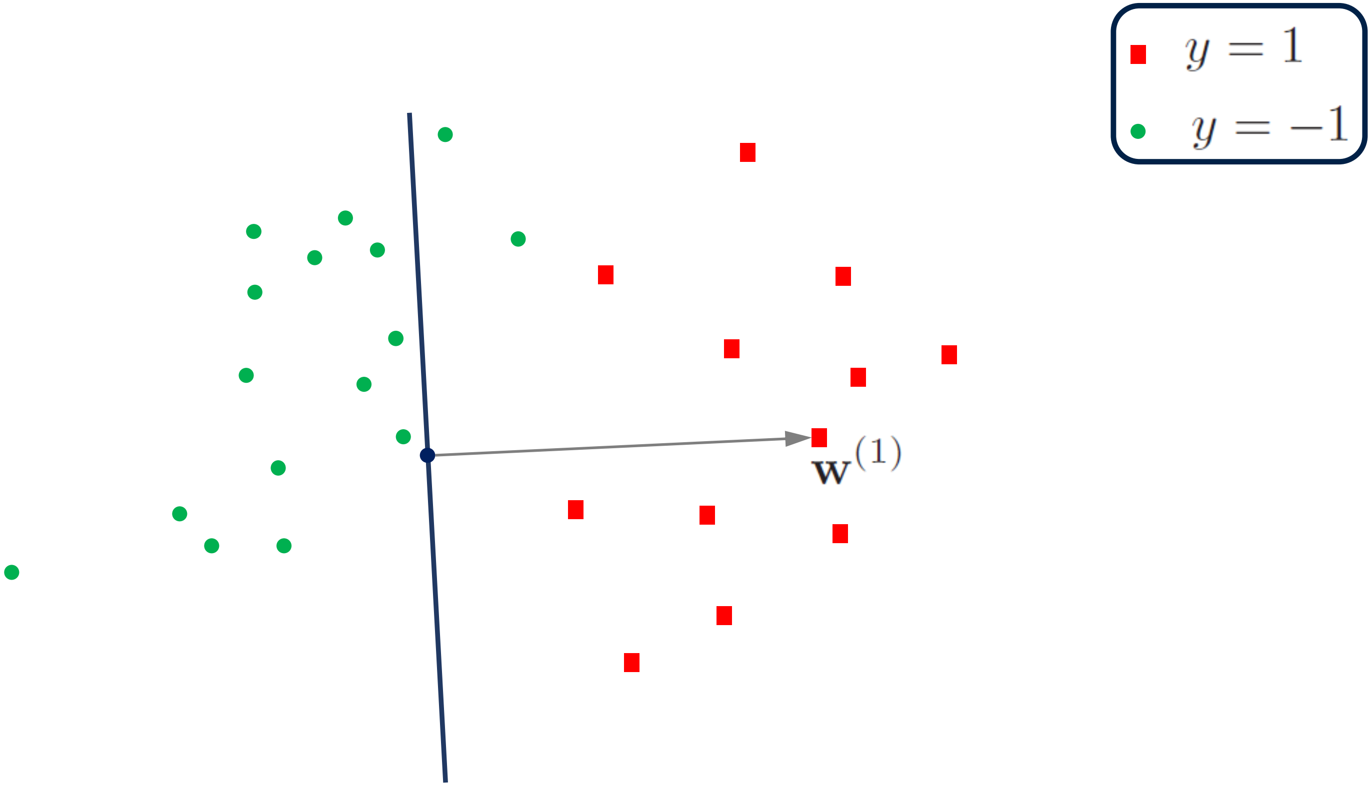
Visualization



Weight update rule:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}$$

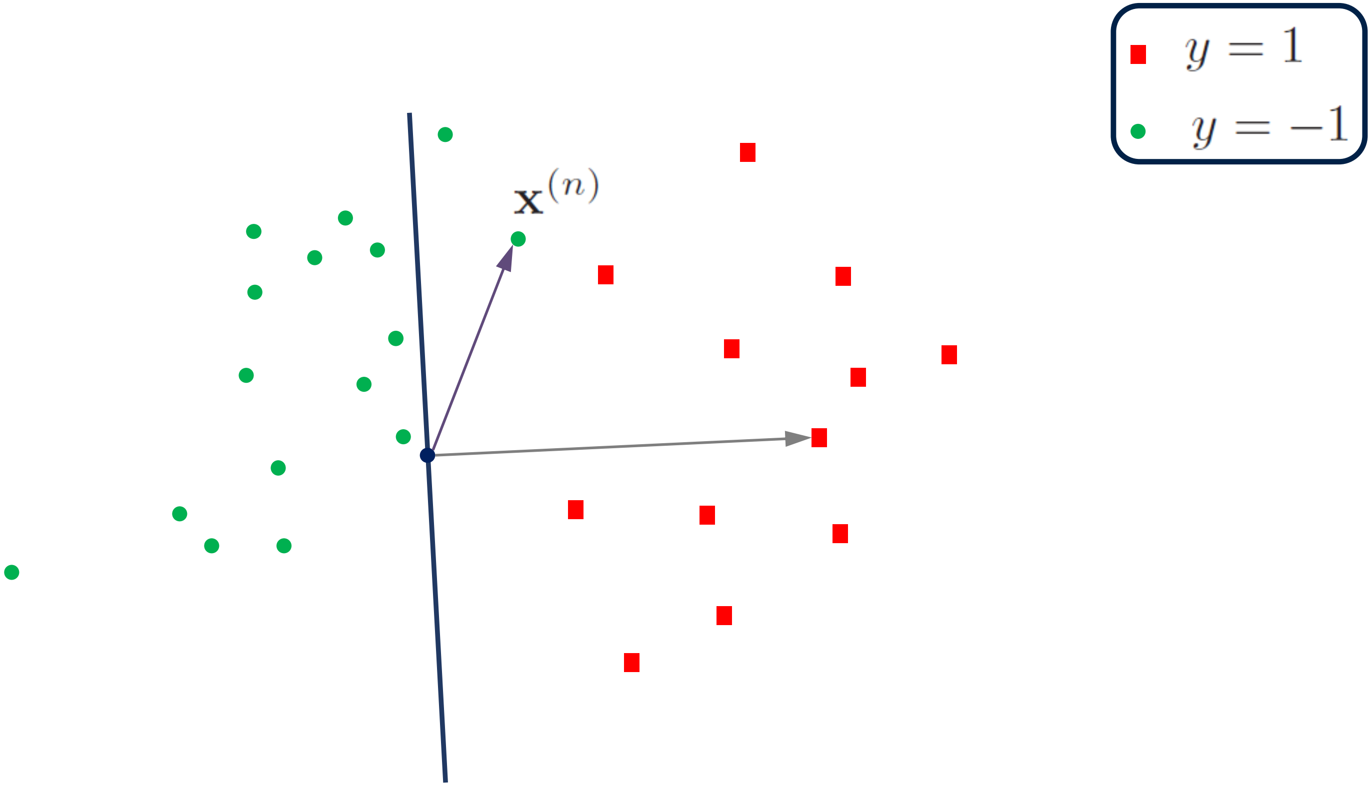
Visualization



Weight update rule:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}$$

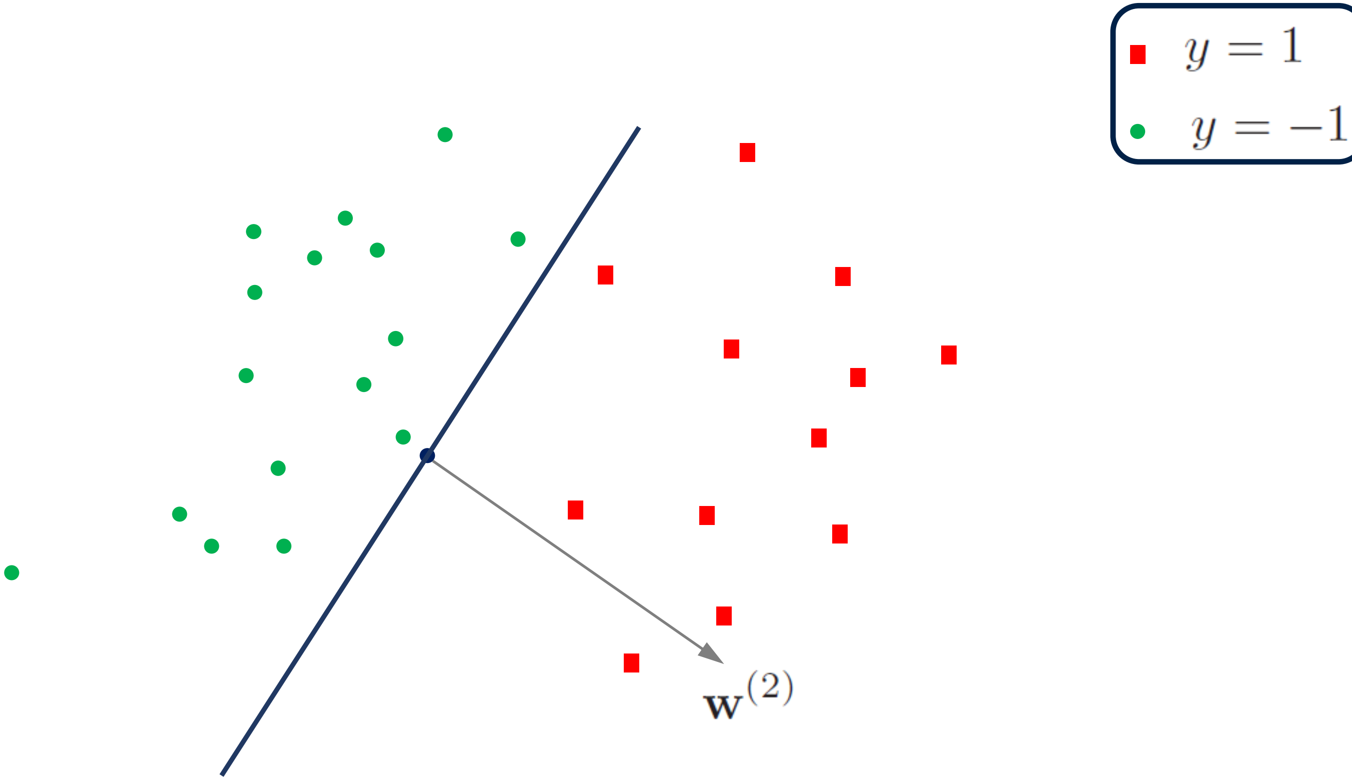
Visualization



Weight update rule:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}$$

Visualization



Weight update rule: $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}$

Convergence Theorem

Theorem: If the training data is linearly separable with margin γ by a unit norm hyperplane $\mathbf{w}^*$ ($||\mathbf{w}^*|| = 1$) with $w_0 = 0$, then perceptron converges after $1/\gamma^2$ mistakes during training (assuming $||\mathbf{x}|| < 1 \;\; \forall \mathbf{x}$).

- Note, for any datatset the inputs $\mathbf{x}$ can be scaled as $\mathbf{x}/\max_{n=1:N} ||\mathbf{x}^{(n)}||$ so as to achieve $||\mathbf{x}|| < 1 \;\; \forall \mathbf{x}$.

- Consider a hyperplane $\mathbf{w}^*$ such that the following conditions hold:
  - $y^{(n)}\left(\mathbf{x}^{(n)}\right)^{\mathrm{T}}\mathbf{w}^* > 0 \; \forall \left(\mathbf{x}^{(n)}, y^{(n)}\right) \in \mathcal{D}$
  - $||\mathbf{w}^*|| = 1$

- Will check the effect of weight update, i.e. $\mathbf{w}$ to $\mathbf{w} + y\mathbf{x}$, on $\mathbf{w}^{\mathrm{T}}\mathbf{w}^*$ and $\mathbf{w}^{\mathrm{T}}\mathbf{w}$.

Convergence Proof

$$
\begin{aligned}
(\mathbf{w}^{(k+1)})^T \mathbf{w}^* &= (\mathbf{w}^{(k)} + y^{(n)} \mathbf{x}^{(n)})^T \mathbf{w}^* \\
&= (\mathbf{w}^{(k)})^T \mathbf{w}^* + y^{(n)} ((\mathbf{x}^{(n)})^T \mathbf{w}^*) \\
&\geq (\mathbf{w}^{(k)})^T \mathbf{w}^* + \gamma \\
&\geq (\mathbf{w}^{(k-1)})^T \mathbf{w}^* + 2\gamma \\
&. \\
&. \\
&\geq k\gamma \\
||(\mathbf{w}^{(k+1)})^T \mathbf{w}^*|| &\geq k\gamma \\
||(\mathbf{w}^{(k+1)})^T|| ||\mathbf{w}^*|| &\geq k\gamma \\
||(\mathbf{w}^{(k+1)})|| &\geq k\gamma \quad (\text{as} \quad ||\mathbf{w}^*|| = 1)
\end{aligned}
$$

- This is a lower bound on the weights.

Convergence Proof

$$
\begin{aligned}
||\mathbf{w}^{(k+1)}||^2 &= ||\mathbf{w}^{(k)} + y^{(n)}\mathbf{x}^{(n)}||^2 \\
&= ||\mathbf{w}^{(k)}||^2 + ||y^{(n)}\mathbf{x}^{(n)}||^2 + 2((\mathbf{w}^{(k)})^T\mathbf{x}^{(n)})y^{(n)} \\
&= ||\mathbf{w}^{(k)}||^2 + ||\mathbf{x}^{(n)}||^2 + 2((\mathbf{w}^{(k)})^T\mathbf{x}^{(n)})y^{(n)} \\
&\leq ||\mathbf{w}^{(k)}||^2 + ||\mathbf{x}^{(n)}||^2 \quad (\text{as } \ \mathbf{x}^{(n)} \text{ is misclassified, so}((\mathbf{w}^{(k)})^T\mathbf{x}^{(n)})y^{(n)} \leq 0) \\
&\leq ||\mathbf{w}^{(k)}||^2 + 1 \quad (\text{as}\, ||\mathbf{x}^{(n)}|| \leq 1) \\
&\leq ||\mathbf{w}^{(k-1)}||^2 + 1 + 1 \\
&\,. \\
&\,. \\
&\leq k
\end{aligned}
$$

- This is an upper bound on the weights.

Convergence Proof

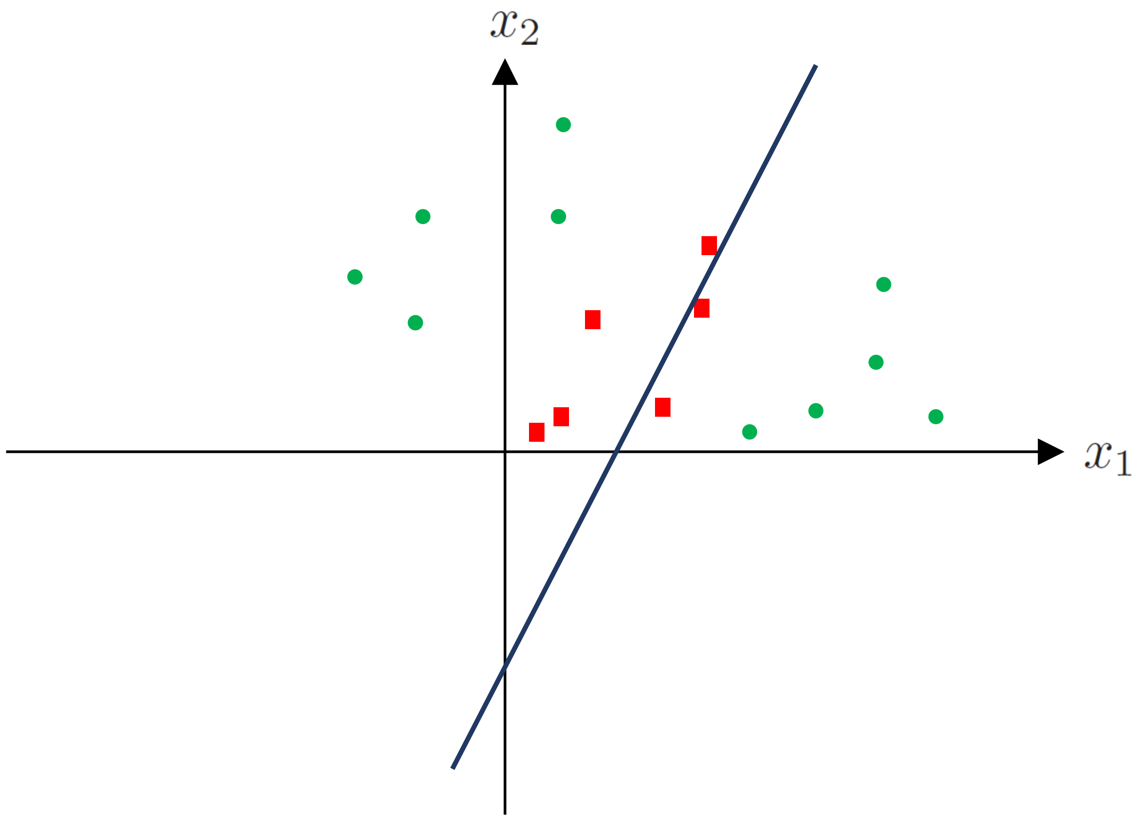
- Using the two inequalities (lower and upper bounds), we have

$$k^2\gamma^2 \leq ||\mathbf{w}^{(k+1)}||^2 \leq k$$

$$\boxed{k \leq \frac{1}{\gamma^2}}$$

- The number of updates k is bounded from above by a constant.
  - The convergence rate is independent of the dimensionality of the dataset D and the number of examples N.

Limitations



XOR function

- XOR data is not linearly separable.

x_1	x_2	XOR	Class label
0	0	0	$\mathcal{C}_2$
0	1	1	$\mathcal{C}_1$
1	0	1	$\mathcal{C}_1$
1	1	0	$\mathcal{C}_2$