Out[2]: Click here to toggle code display on/off.

A very short answer

In compuntational ML what matters for "hardness" of learning is distance to the classification boundary.

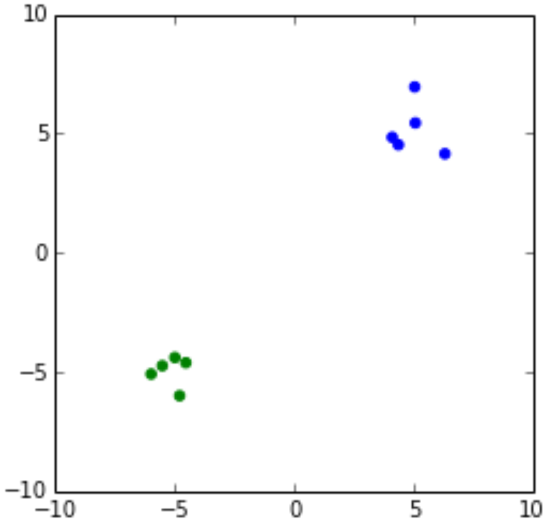
For example: for any hyperplane separating the two classes perfectly (a line in the 2D case) - what could matter is the distance to the sample closest to the hyperplane (line). In modern ML it usually does not matter how the boundary is oriented.

A slightly longer answer

Suppose the two classes come from normal distibutions with mean $\mu_1 = (low, low)$ and $\mu_2 = (high, high)$ and variance $\sigma^2 = 1.0$.

We can visualize this by a picture below:

```
Generated data. Green points are from class 1, blue from class 2
```



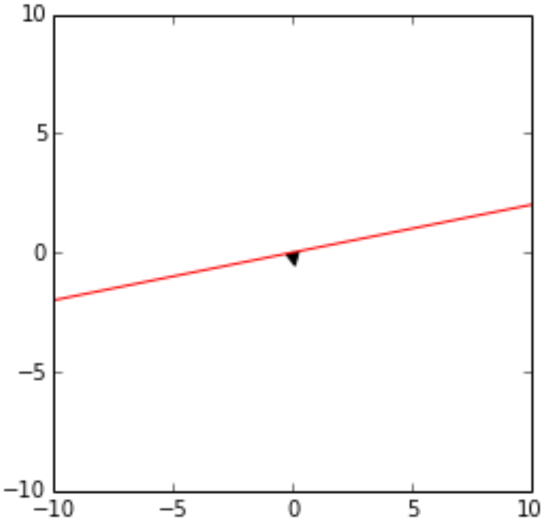
First, suppose we are learning a simple linear classifier. In the case of a very simple perceptron (a neural network with no hidden layers) we will try to learn a matrix W and a weight vector b such that:

$y = Wx_{class_1} + b < 0$ and $y = Wx_{class_2} + b > 0$

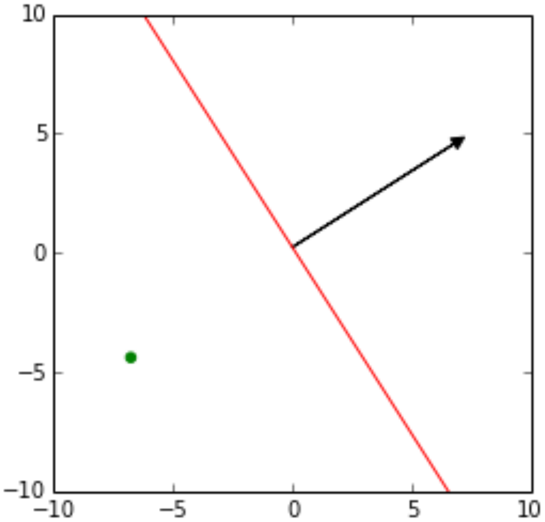
The perceptron learning algorithm adjusts the decision boundary on every iteration when the perceptron makes a mistake on the presented sample.

Below I coded up a very simple visualization that would take the above dataset and iterate on to learn the linear deicision bounday.

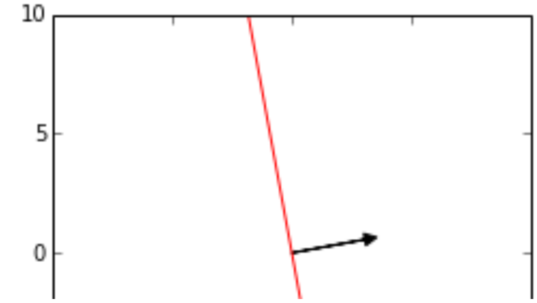
```
Iteration 0
```

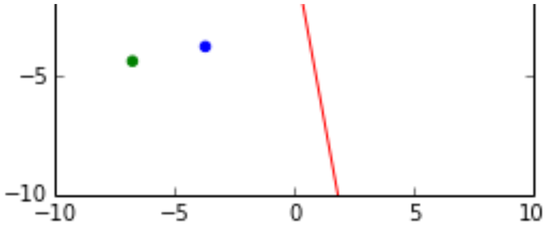


```
Iteration 1
```

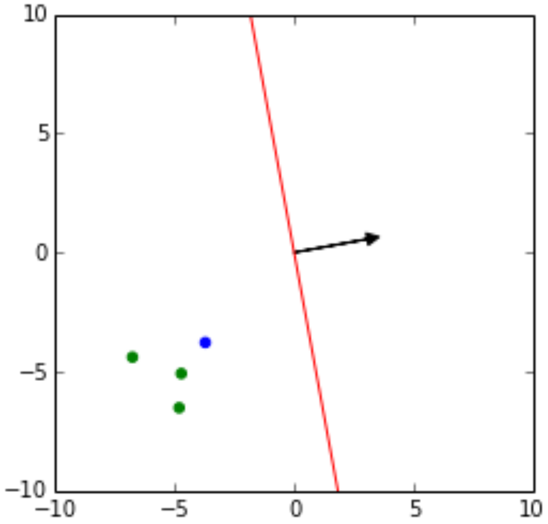


```
Iteration 2
```

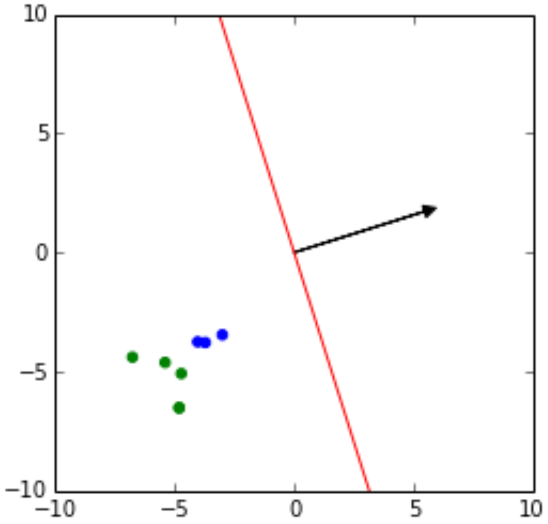




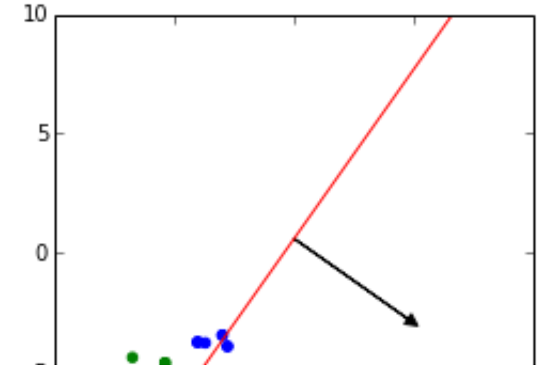
```
Iteration 4
```

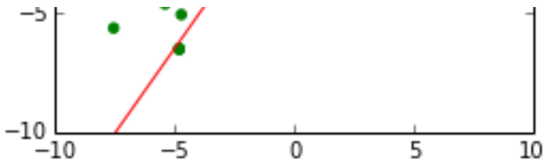


```
Iteration 8
```



```
Iteration 16
```

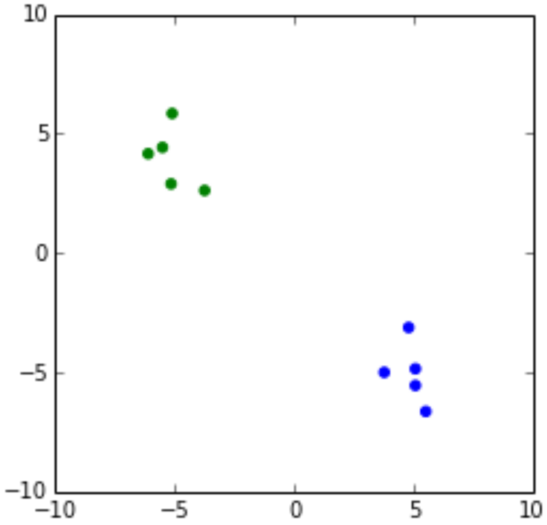




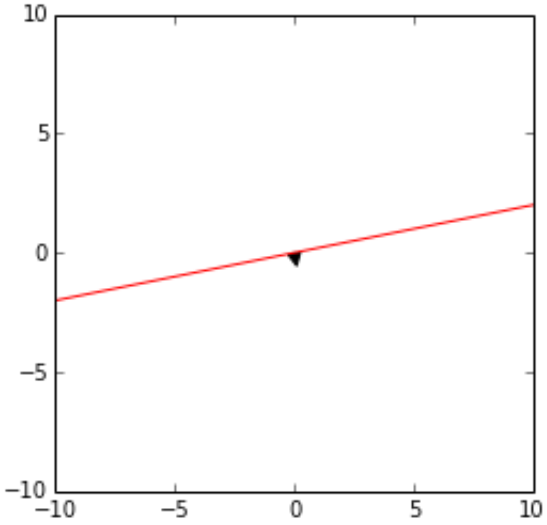
Learning "low-high high-low"

In the above you can see that a good separation is found very quickly. But then the same behavior would be observed for the "low-high high-low" case too!

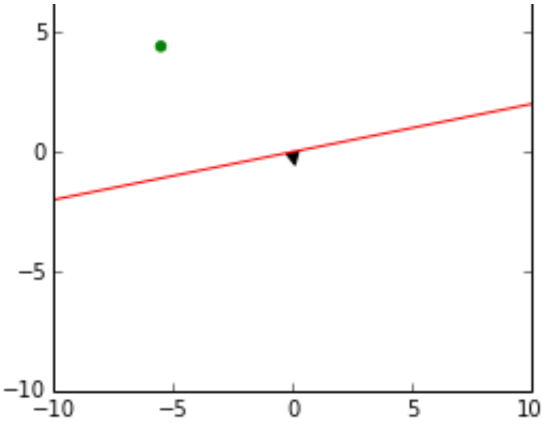
```
Generated data. Green points are from class 1, blue from class 2
```



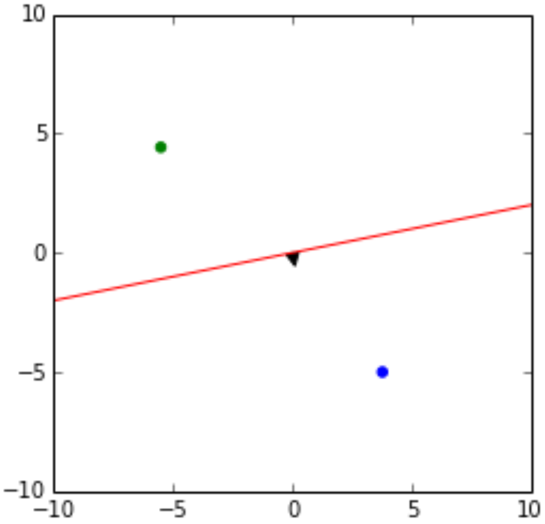
```
Iteration 0
```



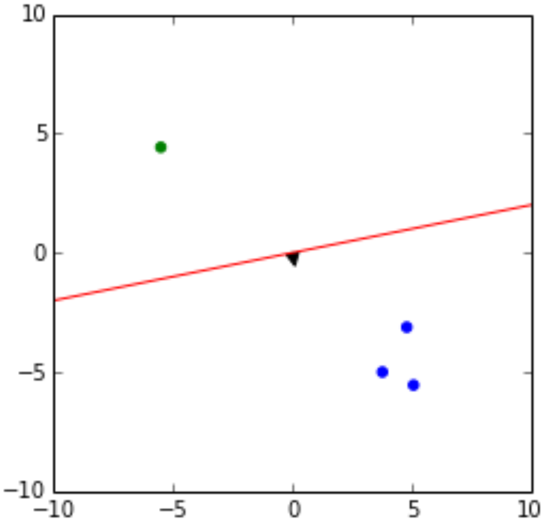
```
Iteration 1
```



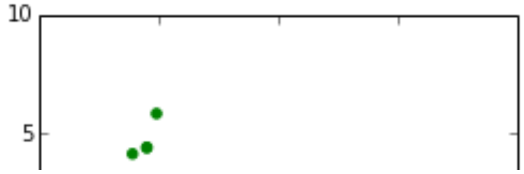
Iteration 2

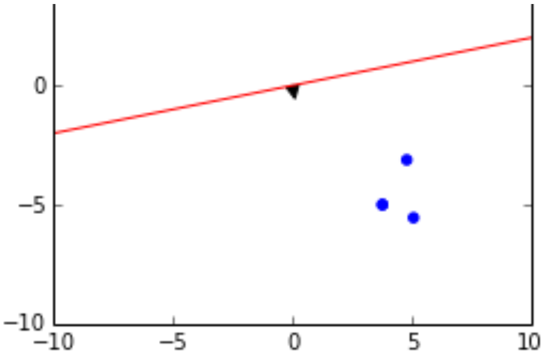


Iteration 4

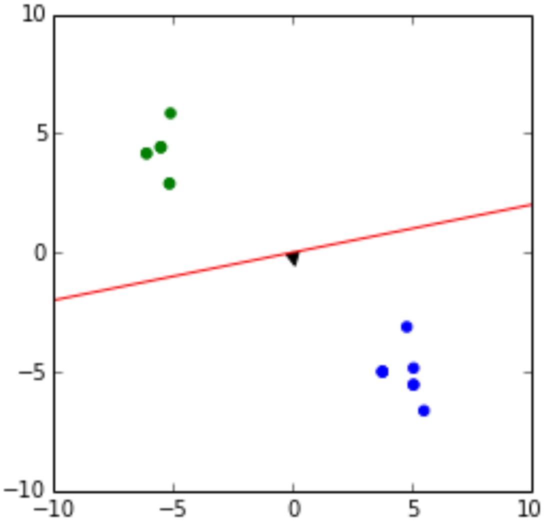


Iteration 8





```
Iteration 16
```

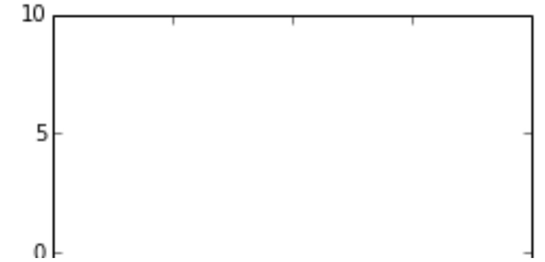


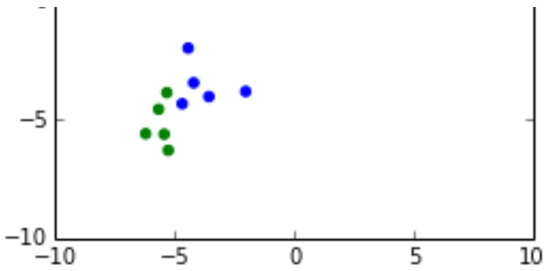
The Symmetry

The issue here is that the problem is symmetric: if you flip one of the axes, then the "low-high high-low" becomes "low-low, high-high" setting. So it is no different from the general learning perspective.

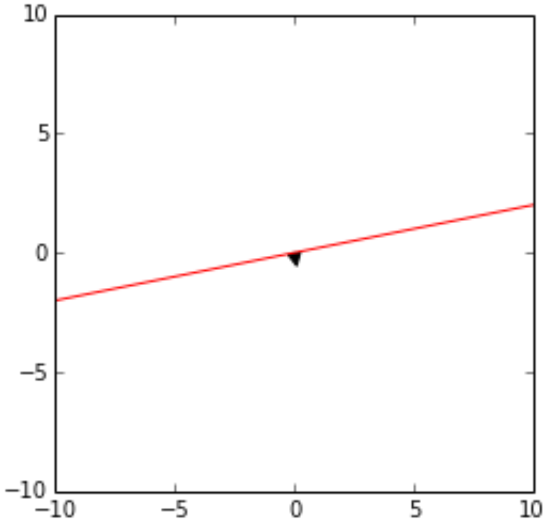
But what would make the setting harder is bringing the two clusters closer in the mean. Suppose that instead of $\mu_2 = (high, high)$, the second mean was and $\mu_2 = (low + 1.0, low + 1.0)$. And now we can see that we are not learning a good decision boundary nearly as fast.

```
Generated data. Green points are from class 1, blue from class 2
```

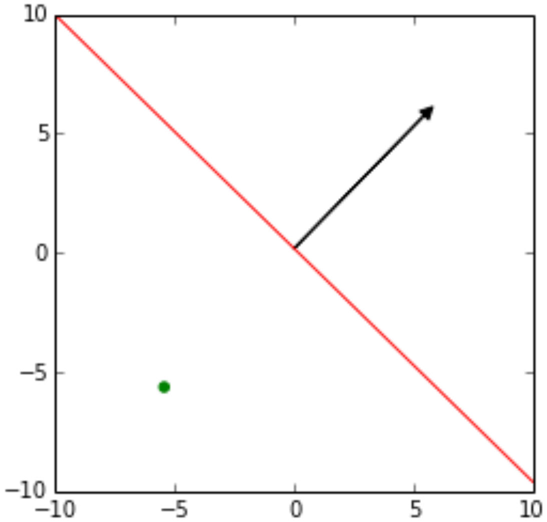




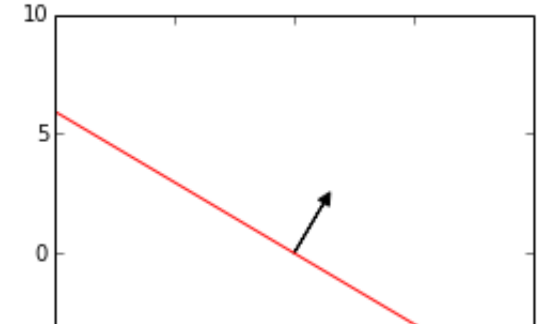
Iteration 0

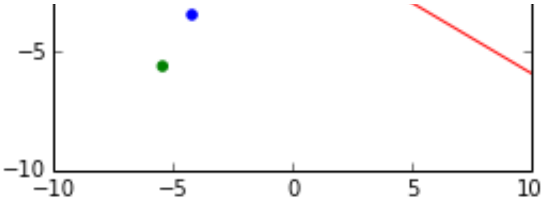


Iteration 1

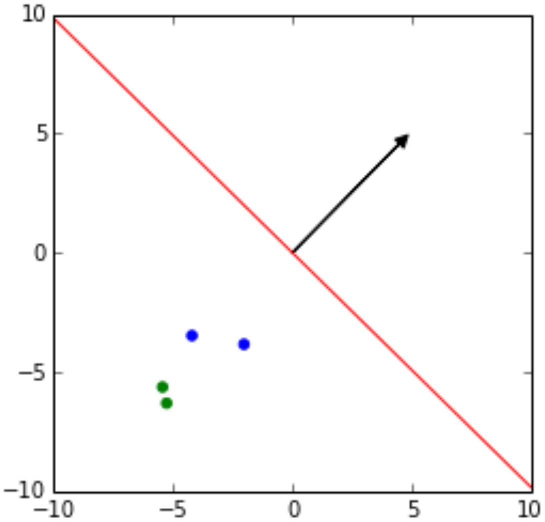


Iteration 2

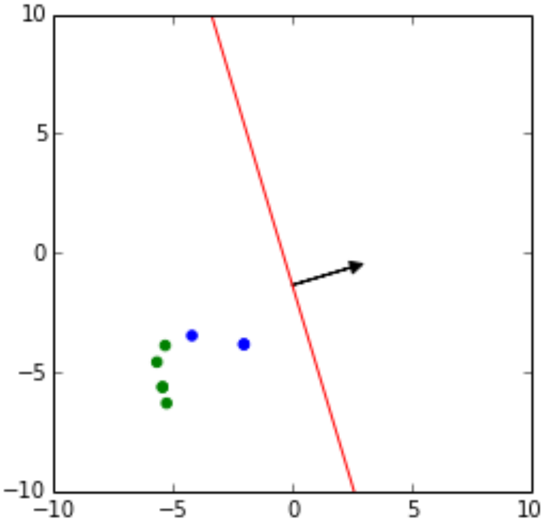




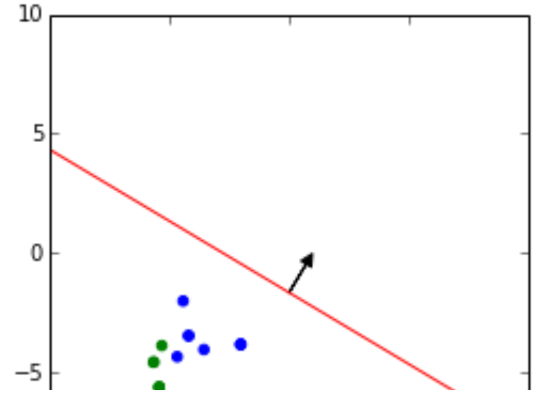
Iteration 4

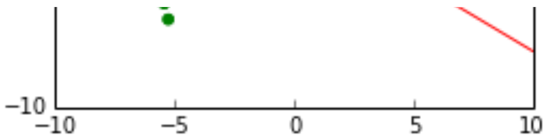


Iteration 8

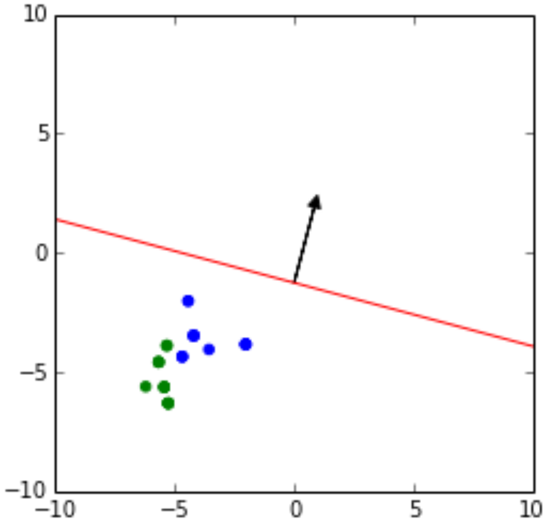


Iteration 16

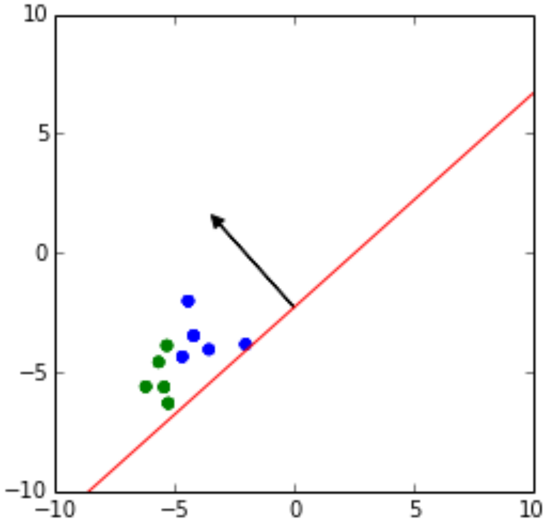




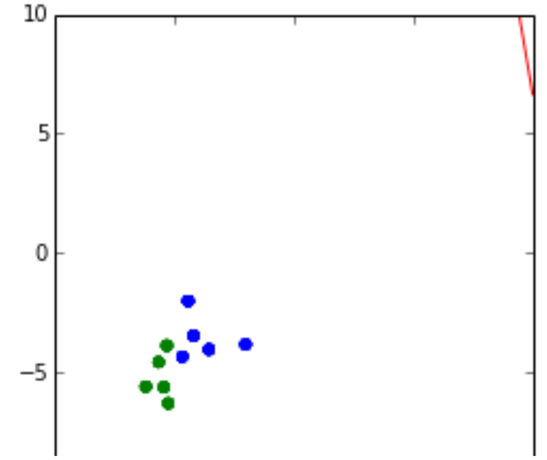
```
Iteration 32
```

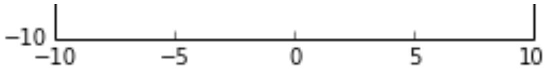


```
Iteration 64
```

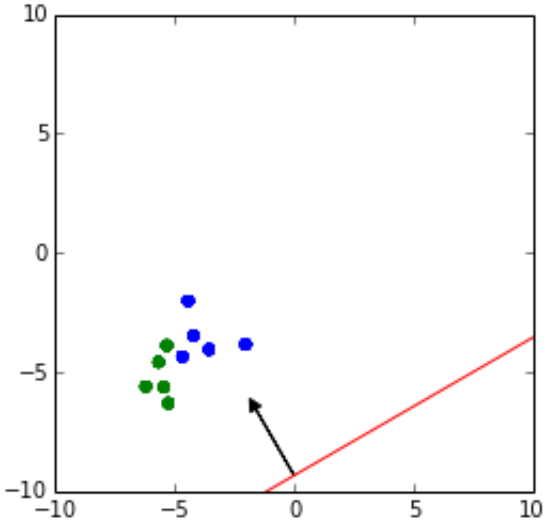


```
Iteration 128
```

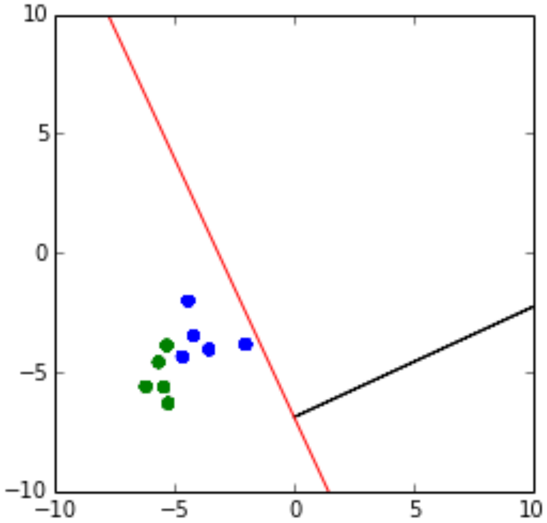




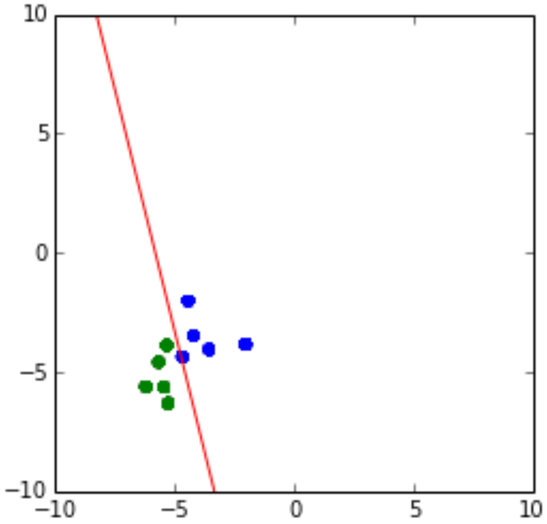
```
Iteration 256
```



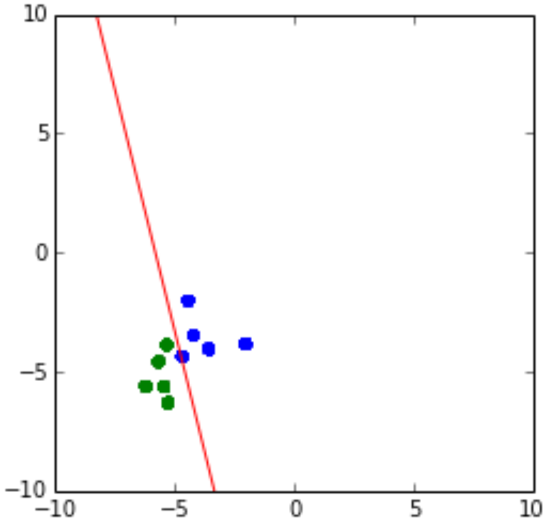
```
Iteration 512
```



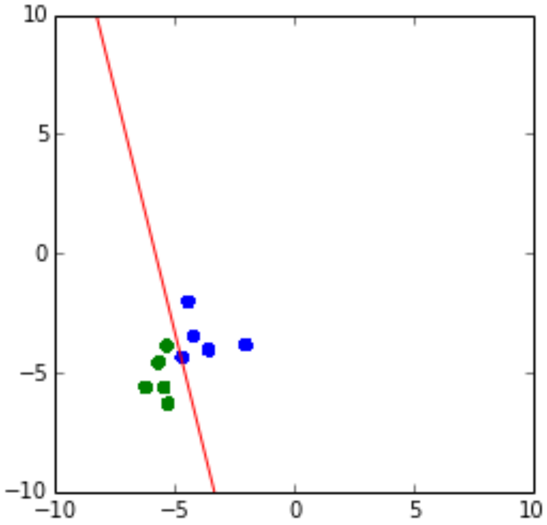
```
Iteration 1024
```



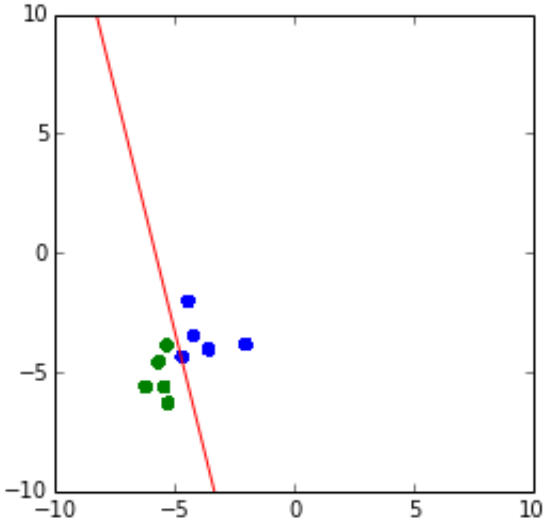
```
Iteration 2048
```



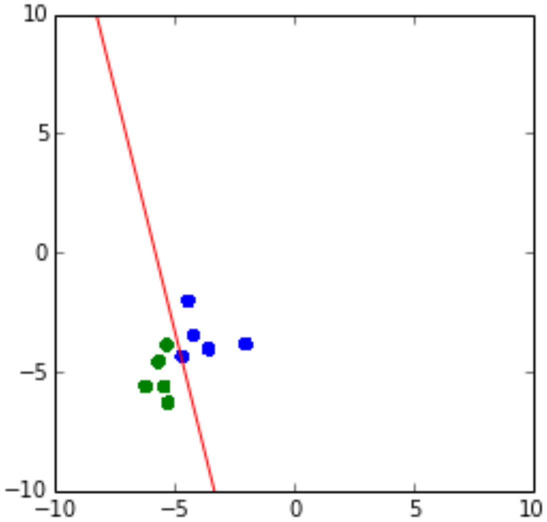
```
Iteration 4096
```



```
Iteration 8192
```



```
Iteration 16384
```



The Hardness

"Harder to classify" can be interpreted in slightly different ways. One of the standard ways is measuring how many errors an algorithm will make before learning the correct separating boundary. And one way to start looking at this is: if we start with random weighs, how many samples will we need to be able to reliably distinguish the data.

But when starting from randomly chosen weights and updating using randomly chosen samples: the two settings are not fundamentally different.

The Intuition

So then where does the intuition of "low-low high-high" being simpler come from? One source could be the fact that it is harder for us to think intuitively of negative numbers: addition is intuitively simpler than subtraction. Indeed, if we are only allowed to add the numbers coming into the clasifier, and then set a separating threshold: the "low-high high-low" setting could be impossible to learn.

So is this intuition wrong? Not always.

In real-world biological systems there might be certain constraints which make addition easier than sibtraction. Such systems might learn using the perceptron rule, but there might be asymmetry in how easy it is to adjust weights in different directions. Are you considering a setting where the system might not be symmetric in its ability to add vs subtract? Maybe something along the lines of stimulation and inhibition being asymmetric? Then the intuition could indeed be applicable.

There could also be a very strong prior that biases the systems to be successful in learning the "low-low high-high" setting faster. For example, if the environment is mostly composed of "low-low high-high" settings, then the organisms living in this environment could be biased to be able to learn such settings better. Could such priors be prevalent in the biological systems you are considering? Perhaps you can think of specific examples/arguments and then use these to show why this intuition could be applicable in your situation.

Generally useful literature

I'll add more here if we have a further discussion, but for now:

One example of an illustrative ML book: "*Pattern Recognition and Machine Learning*" *by Bishop*.

https://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738 (https://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738)

This is one of the best books on intro to ML: sufficiently mathematically rigurous, but also illustrative. A copy of it seems to be available here: http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf (http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf)

An older classic with a bit more intro to learning theory (complexity, mistake bounds, etc): *Tom Mitchell*'s "*Machine Learning*" book:

http://www.cs.cmu.edu/~tom/mlbook.html (http://www.cs.cmu.edu/~tom/mlbook.html)

https://www.amazon.com/Learning-McGraw-Hill-International-Editions-Computer/dp/0071154671 (https://www.amazon.com/Learning-McGraw-Hill-International-Editions-Computer/dp/0071154671)