# DataGenerator

Liniendicke = ?

Wenn liniendicke = 7 (wie bei angleNetwork): Wenn ich will dass immer die volle Breite des Striches im Bild ist, dann ist der Bereich in dem der Strich sich bewegen kann bei imageSize = (29, 29) =>

[0 + 7/2, 29 – 7/2]

## Entschieden: 7

Wie mache ich es, dass prior neuronen mehr impact haben?

Entweder mehr A Neuronen machen, oder ATilde größer machen?

I use same c = 20, but I multiply ATilde with 20!

I tried to train like this, but then priorWeights do not learn the right stuff, it is too strong.

Retry with ATilde \* 10, 7, 5, 3, 1

Vertical lines haben prior 0!

It seems that a Afactor of 5 is too low after all, as even with fixed prior sometimes the network is not sure if the cross is horizontal or vertical.  
Perhaps I should make the lines thinner (does this help?) and let the lines when generating reach the border (do I do this?) so all areas get equal size (do they tho?)

MAYBE! It would be better to subtract the A(t) instead of adding it to the membrane potential

OMG, as prior neurons spike with 50Hz they only spike like every 20 ms!!! That’s way to seldom. I need more prior neurons!!!!

Also ensure that ASpikes are now correctly assigned

# Get prior finally to have an impact

I scraped Afactor as legi said it is too gekünstelt.  
I tried to increase cPrior from 20 to eg 200, but it increases the prior weights by almost nothing.  
I also already increased the prior firing frequency to 200 Hz, but its also way too little still.

Because of that I decided to use more prior neurons. To determine the number of prior neurons I checked how many input neurons are active for a single time step (=20 – 30). One prior neuron that fires with 200 Hz generates on average 200 spikes / 1000 ms = 1/5 spike/ms.   
**200 Hz generate 1/5 spikes per 1 ms**=> 20-30 input spikes vs. 1/5 prior spike …   
Input weights values are up to 1, prior weights up to 4.  
**So the actual impact of the prior is 1/25  
I will try 100 prior neurons (50 of each class) then we will have 25 vs 25 impact.**