Intro: Jakob

(introduce people)

We are going to mostly talk about the 2 SNN architectures for solving the given problem of detecting the direction that a bright spot is moving across sensors positioned across a one dimensional line. But there will also be mentioning a third SNN in the end.  
  
But the first was a very simple architecture based on a direction selective example. It wasn’t good enough at handling repeated inputs and was therefore replaced with our second architecture, which was inspired by the Reichhardt detector.  
  
The added noise and the data that we trained our networks with were all generated by us.

First idea: Vithun (slide 4-8)

The first idea we had was to use the direction selective example from neuronify but in the other way as well. The way the direction selective example works for the right to left detector (The left layer) is by inhibiting when it goes the wrong direction. Every relay neuron gets inhibited before getting an input spike if the light is going the wrong direction but if it goes the correct way the relay neurons get an input spike before inhibition. It works the same for the left to right detector but with swapped directions. Finally both of them have their relay neurons connected to each respective output neuron.

Problems: Johan (Slide 9)

The architecture does not handle higher speeds because when a relay neuron gets an input spike at the same time or just slightly after inhibition the relay spikes anyway. The architecture also does not handle repeating input well, for example if the point light stops moving then it will start repeating the same direction spike even if it is still.

So considering these problems what did we do instead?

Architecture: Johan

This architecture is inspired by the Reichhardt model. We have light input neurons which receive light input. We simulate lateral inhibition by using inhibitory neurons to suppress the output of the local output neurons. We also have extra excitatory neurons that represent the temporal delay so if it is the second node the first node has time to inhibit it before spiking. We also inhibit repetition with the inhibitory neurons near the bottom. So if the left one activates then the right one the right one will be inhibited before it has time to spike from the delay node.

Results: Toste

These are the results of this architecture with back and forth movement.

5 output spikes when moving left to right with 10 input spikes.

4 output spikes afterwards for right to left and left to right except for the last 4 input spikes.

The last 3 inputs are not detected in time.

We can see the output spikes happen about 0.2 s after movement. It is also the reason why there are no spikes for the last little bit of movement. The reason why there are 5 spikes for the first input is because the first input spike is treated as movement but for the others the first input spike is part of the previous directions’s movement.

Discussion: Toste

Because the net uses delay nodes so it takes more time for the spikes to register, which will be a problem if you want to respond to the spikes in real time.

End: Emil