1 Artificial learning

1.1 Plasticity

Neuroplasticity: Ability for the brain to re-organize itself in both structure and function over time due to external and internal events. **Neuroplasticity** is mechanism behind "learning" and is happening continuously.

Structural Plasticity	Functional Plasticity
new neural connections	changing existing connections
long-term changes	short term changes

Plasticity happens on all levels from cortical down to the synaptic level.

- **cortical**: changing stimulus from limbs triggers different existing neurons
- **synaptic**: changing amount of gates on post-synaptic neurons' dendrites

1.2 synaptic strength in functional plasticity

1.2.1 Long Term Potentiation (LTP)

HFS: 100 Pulses (over 1s $\rightarrow$ 100Hz) as an input to a neuron. The neuron is resting at $t = 0$. The **HFS** hits the neuron resulting in an instantaneous output, the **LTP**. The neurons output jumps, then receedes and continues to saturate (Only as long as the **HFS** is continous.) The **synaptic strength** is the chance the output is increased.

A lot of fast input $\rightarrow$ Big changes and high learning

LTP increases synaptic strength

1.2.2 Long Term Depressino (LTD)

The Inverse, to decrease the **synaptic strength** an **LFS** (900 Pulses 15min $\rightarrow$ 1Hz) is sent. The neuron responsds, dips and saturates in a depression.

Low data $\rightarrow$ Low learning

LTD decreases synaptic stregth

1.2.3 Chemical basis

LTP and **LTD** result in synapeses by creating or destroying gates at the pos-synaptic terminal respectively.

1.3 Hebbian learning model

Efficiency describes the likelyhood if a presynaptic neuron spiking and exciting it's postsynaptic neuron. The likelyhood of the post-synaptic neuron firing after having been exicted is increased. More firing together $\rightarrow$ more likely to fire together in the future. They spiking is, however, not necessarily causal. At high efficiency the spiking of both neurons are **temporally correlated**. The spiking is **associative** and **unsupervised**.

Neurons that fire together, wire together.

1.3.1 Simple mathematical model

$$\frac{d\omega_1}{dt} = \mu \cdot v \cdot u_1$$

- ω: dsecribes the synaptic strength / weight
- $\frac{d\omega_1}{dt}$: (not a derivative), Change in synaptic weight
- μ: Learnig rate ($\mu \ll 1$ to avoid "exploding learning problem")
- v: Output of post-synaptic neuron
- u_1: Output of pre-synaptic neuron / input to post-synaptic neuron

$$\omega_n = \omega_{n-1} + \frac{d\omega_{n-1}}{dt} = \omega_{n-1} + \mu \cdot v \cdot u_{n-1}$$

Problem: ω_1 is always increasing, unstable ~~but biologically correct~~. This is an open control loop.

As this is unsupervised we don't have an error term and can't simply stop when the model is "good enough".

1.3.2 LTP

The further the amount of time between two spikes firing the more the weight changes. A high δt results in little change, a small δt results in large changes. At $\delta t = 0$ maximal change occurs. The simple model only results in positive change, thus unstable.

1.4 Input correlation learning (ICO)

Learning rule

$$\frac{\delta w_a}{\delta t} = \eta \cdot f(A,t) \otimes \frac{\delta f(B,t)}{\delta t}$$

- η: learning rate
- $f(A,t) \otimes \frac{\delta f(B,t)}{\delta t}$: Temporal correlation
- $otimes$: cross correlation
- A: Predictive signal
- B: Reflex signal
- Y: Neuron Ouput
- w_a weight between A and Y
- f output function of a neuron (including the sigmoid)

If we'd like to stop the learning we can assume B to be constant. We cannot guarantee $B \to 0$ (to stop learning) but we can take the derivative to stop learning once stimulus ceases change.

This Algorithm **will converge** to the correct weight.

Output signal is the weighted sum.

$$Y(t) = w_a \cdot f(A,t) + f(B,t)$$

1.4.1 Perceptron learning

Learning by updating input weights only. Update done using **gradient descent**.

Update weight in proportion to contribution to the output. Contribution is the change in error E für a given change in w, where the mean squared error is defined as

$$E = \frac{1}{2}(t-v)^2$$

- t: target output
- v: actual output

Determining error requires a known correct output.

$\rightarrow$ **supervised learning**

Derivative of error is a gradient of E. Finding the global minimum is done using gradient descent. The error gradient for a sigmoid is given by

$$\begin{aligned}\frac{dE}{dw} &= \frac{1}{2} \cdot 2\frac{d(t-v)}{dw}(t-v) \\ &= v(1-v)(t-v)\end{aligned}$$

Updates on weights are done using

$$\frac{dw_i}{dt} = \mu \cdot v(1-v)(t-v) \cdot u_i$$