*KRCP Lecture 6*

*Learning*

**Types of learning in mammals**

* **Associative Learning**: Classical Conditioning and Operant Condition (modify occurrence of behavior)
* **Non-associative Learning**: Habituation (decrease in response to repeated stimulus) and Sensitization (amplification of response to repeated stimulus)

**Learning over different time scales**

* **Development learning**: learning that takes place as a normal part of cognitive development
* **Behavioral learning**: observable change in behavior
* **Evolutionary learning**: learning that takes multiple generations

**Hebbian learning**

* “Cells that wire together fire together”
* Cells that fire together strengthen synapses
* Cells that fire separately lose synapses

**Long-term potentiation/depression (LTP/LTD)**

* Molecular mechanism for (Hebbian) learning
* **LTP**: (E)PSP is increased after presynaptic stimulation 🡪 Long-term synaptic strengthening
* **LTD**: (E)PSP is decreased after presynaptic stimulation 🡪 Long-term synaptic weakening
* **Associative LTP**: Implementation and Extensions of Hebb’s law, 3 properties: **Cooperativity** (works when more than 1 input is simultaneously active), **Associativity** (weak inputs are potentiated when occurring with string inputs) and **Specificity** (only stimulated synapse shows potentiation)
* NMDA receptor: Central to producing LTP

1. Glutamate binds on receptors
2. Post-synaptic cell is depolarized
3. Mg2+ ejected out of channel
4. Ca2+ enters and facilitates LTP

**3 projection pathways in Hippocampus**

* **Perforant pathway** synapses in the dentate gyrus
* **Mossy fiber pathway** synapses in CA3
* **Schaffer collateral pathway** synapses in CA1

**Types of Machine Learning**

* **Supervised Learning**: teacher
* **Reinforcement Learning**: a bit of help (rewards)
* **Unsupervised learning**: all on your own (Content addressable memories)

**Artificial Neural Networks**

* **Units/Nodes**: have input/output and perform simple computations
* **Connection weights**: synaptic strength between 2 nodes
* **Activation function**: transforms weighted sum into an activation value, 3 types: step, sign and sigmoid function
* **Layers**: single ANN has two, input and output, and multilayer ANNs have hidden layers between those two
* **Bias node**: constant output of 1 or -1
* **Types of general structure**: **fully connected** (all units to all), **feed-forward** (one layer to next) and **recurrent-network** (some back connection)
* **Perceptron**
  + Most simple form of an ANN
  + **Weight learning rule**: learning constant \* error \* output
  + Perceptron can only classify **linear separable** problems
  + Multi layer perceptron:
    - Add one or multiple **hidden layers**
    - Needs a **differentiable activation function**
    - Back-propagation rule (weight change): learning constant \* error
    - Pro: easily parallelizable, can calculate anything
    - Con: Learning takes a long time and depends on initial weights, black-box, cannot find global optimal solution (only local minimum)