Learning frameworks

**Supervised learning**
- Assumes environment specifies correct output (targets) for each input

**Unsupervised learning**
- Assumes environment only provides input; learning is based on capturing the statistical structure of that input (efficient coding)

**Reinforcement learning**
- Assumes environment provides evaluative feedback on actions (how good or bad was the outcome) but not what the correct/best action would have been

**Associative reinforcement learning**
- Given input, learn to produce output (action) that maximizes immediate reward
- Modified Associative reward-penalty ($A_{R-P}$)
  \[
  p(a_j = 1) = \frac{1}{1 + \exp(-n_j)}
  \]
  \[
  \Delta w_{ij} = \begin{cases} 
  \rho(a_j - n_j) a_i & \text{if success} \\
  \lambda \rho((1 - a_j) - n_j) a_i & \text{if failure} 
  \end{cases}
  \]
- Reinforcement is broadcast within multilayer network

**Adaptive critic**
- Feedback can be intermittent, probabilistic, temporally delayed

Optimal/effective actions are not provided to learner; must be discovered
- Feedback (reinforcement signal) reflects overall consequences of action (and other things) in environment
- Feedback can be intermittent, probabilistic, temporally delayed, and dependent on things outside learner’s control
- Tension between exploration and exploitation
Sequential reinforcement learning

- Execute sequence of actions that maximizes expected discounted sum of future rewards

\[ E \{ r(t) + \gamma r(t+1) + \gamma^2 r(t+2) + \cdots \} = E \left\{ \sum_{k=0}^{\infty} \gamma^k r(t+k) \right\} \]

- Temporal difference (TD) methods
  - Learn to predict expected discounted reward

\[
\begin{align*}
  a_j(t+1) &= E \{ r(t+1) + \gamma r(t+2) + \gamma^2 r(t+3) + \cdots \} \\
  a_j(t) &= E \{ r(t) + \gamma r(t+1) + \gamma^2 r(t+2) + \gamma^3 r(t+3) + \cdots \} \\
  &= E \{ r(t) \} + \gamma a_j(t+1) \\
  E \{ r(t) \} &= a_j(t) - \gamma a_j(t+1) \\
  \Delta w_j(t) &= \rho(r(t) - E \{ r(t) \}) a_i \\
  &= \rho(r(t) - (a_j(t) - \gamma a_j(t+1))) a_i
\end{align*}
\]

- Use as internal reinforcement for learning actions

---

Dopamine and reward prediction (Shultz et al., 1997)

- Classical conditioning
- Response of dopaminergic neurons in **substantia nigra** (subcortical nucleus)
Strengths and limitations of reinforcement learning

**Strengths**
- No need for explicit behavioral targets
- Can be applied to networks of binary stochastic units
- TD learning consistent with some physiological evidence (Schultz)
- Can use associative reinforcement learning (e.g., $A_{R,P}$) to learn actions based on prediction of reinforcement learned by TD

**Limitations**
- Learning is often very slow (not enough information)
- Application to large/continuous state spaces requires some mechanism for function approximation (e.g., multilayer back-propagation network; deep reinforcement learning)
- Associative and TD learning combined only in very simple domains (but deep learning can also be applied to state representations; e.g., auto-encoder)

**Forward models**
- Feedback from the world is in terms of *distal error* (observable consequences) rather than *proximal error* (motor commands)
- Would like compute proximal error from distal error (to improve motor commands to achieve goals)
- Relationship between motor commands and observable consequences involves processes in the external world (e.g., physics)
- Learn an *internal (forward) model of the world* which can be *inverted* (e.g., back-propagated through) to convert distal error to proximal error
  - Such a model can also provide online outcome prediction to detect errors during execution
Training

- **Forward model: predicted – actual**
  - Generate action randomly, predict outcome
  - Use discrepancy between predicted outcome and actual outcome as error signal

- **Inverse (action) model: desired – actual**
  - Generate action from “intention” in current context
  - Use discrepancy between generated outcome to actual outcome as error signal
  - Back-propagate error through forward model to derive error derivatives for action representation
  - Back-propagate action error to improve inverse model

- Forward and inverse models can be trained at the same time
Figure 20. The workspace (the gray region) and four target paths: The trajectories move from left to right along the paths shown.