Actualmente se está trabajando con modelos de mundo que aprenden en tiempo real a predecir acciones que en un futuro les puede generar una pérdida o una ganancia. Llámese pérdida a las acciones que estarían mal hechas, como por ejemplo: tener como objetivo agarrar una bola, pero que la acción decidida por el robot sea simplemente golpearla; una acción que le produzca ganancia sería agarrar la bola (es el objetivo inicial).

Las acciones que se pueden ejecutar por el robot hasta ahora son las siguientes:

\* Agarrar objeto

\* Pedir favor (Ej: Pedir que acerquen una caja)

\* Cambiar objeto de mano

\* Poner objeto en una caja

\* Arrastrar objeto

\* Agarrar con ambas manos

\* Tirar

\* Poner el objeto en el robot

El objetivo es poder aprender basándose en los resultados que me dieron las acciones que he ejecutado anteriormente. Es decir, saber que si mi objetivo era agarrar la bola y lo que hice fue golpearla, aprender que esa acción para este objetivo, no me puede traer ningún beneficio entonces tiene una probabilidad baja de ejecución, mientras que agarrar la bola tiene una probabilidad alta pues anteriormente me dio un buen resultado.

Cada acción que se quiere predecir va a depender completamente del contexto actual en el que se encuentre el robot. Porque el ambiente es muy poco probable que sea idéntico a los anteriores, es por esto que las predicciones son probabilidades ([0, 1] probabilidad de ejecutar o no una acción).

La idea es que por cada acción (llámese acción a un movimiento que pueda ejecutar el robot como mover una mano) haya un mecanismo de aprendizaje automático que pueda decidir qué tan probable es que sea una buena para lograr un objetivo. Podemos decir que un objetivo puede componerse de una o más acciones, es por esto que las anteriores dependen de las siguientes, y se debe de aprender de estas, es decir: olvidar lo menos posible.

Dentro de la memoria de largo plazo del MDB existen modelos de aprendizaje automático cuya función es predecir si ejecutar o no cierta acción (como mover alguna parte, o puede ser incluso un grupo de acciones); estos son llamados pnodes. Los pnodes reciben información del contexto actual y predicen la probabilidad de que la acción que les corresponde sea buena o mala para el objetivo propuesto. Ejemplo: Para agarrar una bola que se encuentra al lado derecho del robot, es necesario activar los pnodes que tienen como objetivo predecir si: mover el hombro derecho, mover la mano derecha, agarrar un objeto; van a generar alguna recompensa. Para poder realizar este trabajo, cada pnode recibe la siguiente información:

\* ball\_in\_right\_hand: Booleano que indica si el robot tiene la bola agarrada con la mano derecha, en metros.

\* ball\_dist: Distancia que hay entre el robot y la bola, en metros.

\* box\_size: Tamaño de la caja, en metros.

\* ball\_in\_left\_hand: Booleano que indica si el robot tiene la bola agarrada con la mano izquierda.

\* box\_ang: Ángulo del robot a la caja, en radianes de -pi a pi.

\* ball\_ang: Ángulo del robot a la bola, en radianes de -pi a pi.

\*box\_dist: Distancia que hay entre el robot y la caja, en metros.

\*ball\_size: Tamaño de la bola, en metros.

Como se puede observar, los objetivos del robot para estos pnodes serían meter la bola en una caja, mover una caja, agarrar una bola, etc. Hasta el momento, se ha trabajado con 21 pnodes que cada uno ejecuta una acción de las mencionadas anteriormente.

Es importante saber que el aprendizaje del robot es en tiempo real, los datos van ingresando 1 por 1 al modelo de aprendizaje automático, y se necesita algún mecanismo que ayude a aprender el modelo matemático del problema.

Para poder retornar buenas acciones, lo que se hace es: hacer una predicción, y posteriormente recibir una respuesta indicando si la acción realizada estuvo buena o mala. De esta manera, entrenar el modelo de predicciones (cada pnode que participó en la acción) para ajustarlo y aprender de la acción realizada.

El tiempo que le toma al robot realizar una acción puede ser bastante largo (1 min, por ejemplo), por ello lo que se hizo fue simular la ejecución de acciones por parte del robot, y almacenar los datos obtenidos para poder entrenar un modelo de aprendizaje automático. De esta manera se obtuvo un dataset por cada pnode para poder entrenar y probar los modelos. (Cada dataset con datos entre 250 y 1000 aproximadamente)

Hasta ahora, se ha estado utilizando un algoritmo evolutivo llamado Neat como el modelo de apredizaje automático (cada pnode). El algoritmo lo que hace es buscar una topología de red neuronal óptima para la predicción de los datos. Lo que se quiere hacer es buscar un método de Deep Learning que mejore el resultado de las predicciones, olvide muy poco para no perder información y que cada vez que entrene le tome poco tiempo; pues es un trabajo que realiza en tiempo real (como se mencionó anteriormente).