Bio-Inspired Deep Reinforcement Learning for Autonomous Navigation of Artificial Agents

H. Lehnert, M. Araya, R. Carrasco-Davis, and M. Escobar, *Member, IEEE*

***Abstract*—Autonomous navigation of artificial agents is a challenging task for changing and complex environments. Reinforcement learning (RL) algorithms are widely used for autonomous navigation, where the agent, through the interaction with the environment, learns the behaviors needed to maximize the reward. Recent architectures extract information from the environment using convolutional neural networks, where the visual features needed to maximize the reward are unknown and uncertain, and then, increasing the number of parameters learned by the entire system. Moreover, the presence of sparse rewards complicates, even more, the task generating unstable results in the learning problem. The work here presented is twofold. First, we show the advantages of using retina physiology knowledge to design a visual sensor feeding the RL network. Secondly, based on intrinsic motivation, we propose the use of auxiliary tasks to deal with sparse rewards, generating a continuous learning process. We define two auxiliary tasks, state, and action predictions, forcing the network to learn characteristics of environment; and also, to detect which of them are valuable for the task. These two contributions were implemented in the DeepMind Lab environment simulating an agent moving inside two different maze scenarios. The results obtained reveal a promising extension of the inclusion of biological-plausible mechanisms inside artificial intelligence applications. Moreover, to include auxiliary tasks improves the performance adding robustness to the system.**

***Index Terms*—Reinforcement Learning, Autonomous Navigation, Visual Models.**

I. Introduction

EL aprendizaje reforzado (AR) corresponde al área del aprendizaje de máquina donde un agente toma una decisión en cada instante, con el fin de maximizar una recompensa acumulada en el tiempo. La dificultad radica principalmente en la incertidumbre con respecto al ambiente, por lo que es necesario *aprender* mediante el ensayo y error, *reforzando* el comportamiento mediante las señales de recompensa y castigo. Una limitación importante del AR es que, algunas veces, la exploración no sea capaz de captar la complejidad del ambiente afectando el desempeño del agente.

Por otro lado, el acelerado avance de las técnicas de *Aprendizaje Profundo* ha dado un nuevo impulso a esta área. El *Aprendizaje Reforzado Profundo* [1] aprovecha las capacidades de las redes neuronales de extraer representaciones significativas a partir de datos de alta dimensionalidad, y así lograr atacar problemas de mayor complejidad. Por ejemplo, estas técnicas han logrado superar a jugadores expertos en el juego de Go [2], y en particular aprovechar el auge de las redes neuronales convolucionales para extraer de información directamente desde imágenes para aprender a jugar juegos de Atari [3].

Sin embargo, las aplicaciones que interactúan con el mundo real aún presentan importantes desafíos para los algoritmos de aprendizaje reforzado. Sobre todo, porque las fuentes de información no son variadas ni abundantes, sino que redundantes e irrelevantes para la tarea a realizar. Más aún, las tareas pueden presentar recompensas escasas limitando la oportunidad de aprendizaje. En resumen, se requieren mecanismos que guíen el aprendizaje tanto a nivel perceptual como conductual.

En este trabajo se presentan dos mecanismos bio-inspirados, que buscan atacar los problemas anteriormente mencionados, al ser incorporados en un esquema de aprendizaje reforzado.

En primer lugar, se propone el uso de filtros basados en los modelos de células retinales de una especie de roedores [4], de los cuales se ha mostrado que logran resaltar características de interés en un flujo visual. Con estos filtros se realiza un preprocesamiento de los estímulos visuales que entran al sistema de aprendizaje reforzado, simplificando el modelo que debe ser entrenado por el algoritmo. Si bien este **primer mecanismo** fue recientemente presentado en [5], este trabajo profundiza el análisis de estabilidad de las soluciones mediante la selección de tamaños de batch a utilizar. El **segundo mecanismo** busca atacar el problema del uso eficiente de las experiencias y la dificultad en el aprendizaje cuando la recompensa recibida es escasa. Para abordar estos problemas, se proponen mecanismos que toman su inspiración del concepto de motivación intrínseca [6], según el cual el comportamiento de un individuo es impulsado por aquellas cosas que le parecen interesantes y no necesariamente por una necesidad vital o estimulo externo como puede ser la recompensa en el aprendizaje reforzado. Basados en esta idea, se propone el uso de tareas auxiliares de predicción que buscan que el modelo entrenado, además de realizar la tarea que les fue designada, pueda predecir el siguiente estado y acción. En otras palabras, el agente intentará estimar como cambia lo que observa en base a sus acciones, y viceversa, cual será la siguiente acción en base a como cambia lo que observa. Con esto se busca que se puedan aprender características importantes del ambiente incluso en aquellos momentos en que no se recibe recompensa.

Para probar los mecanismos propuestos, se entrenan agentes para realizar la tarea de navegación, la cual consiste en ser capaz localizarse y desplazarse autónomamente por el espacio. Esta es una tarea esencial tanto para agentes biológicos, robóticos o virtuales, por lo que ha sido utilizada en múltiples trabajos que buscan desarrollar nuevos métodos de aprendizaje

H. Lehnert, M. Araya, R. Carrasco-Davis, and M. Escobar, Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile.

R. Carrasco-Davis, Department of Electrical Engineering, Universidad de Chile, Santiago, Chile.

*Email: mariajose.escobar@usm.cl

reforzado [7]–[10].

Para la generación de las experiencias necesarias para el entrenamiento se emplean ambientes virtuales que presentan espacios tridimensionales ricos en características visuales, desde una perspectiva en primera persona. Los modelos entrenados son evaluados en función de su desempeño en comparación con modelos utilizados comúnmente como base de comparación en otros trabajos del estado del arte.

II. Background

A. Aprendizaje Reforzado

Los métodos tradicionales de aprendizaje reforzado no escalan bien a problemas con espacios de estados o acciones de alta dimensionalidad o continuos, debido a la explosión combinatorial de la memoria en representaciones tabulares de las funciones de transición y/o valor. Se hace necesario entonces introducir aproximadores que aprendan estas funciones y generalizar así a estados nunca antes visitados. El uso de redes neuronales para este objetivo se remonta a el exitoso TD-Gammon [11], sin embargo, las técnicas utilizadas en éste y otros trabajos tempranos no resultaron escalables a problemas de mayor dimensionalidad [1].

Uno de los primeros trabajos que lograron surcar los problemas de inestabilidad en el entrenamiento y alta dimensionalidad fue la propuesta de Deep Q-Learning (DQN) [3]. Posterior a este trabajo muchos otros métodos se han propuesto que realizan igualmente de manera satisfactoria y consistente el entrenamiento, empleando otros enfoques, como estrategias evolutivas [12], el uso de arquitecturas híbridas de redes neuronales y algoritmos de búsqueda clásicos [2], y el entrenamiento distribuido de redes [13], entre otros.

B. Aprendizaje Reforzado para Navegación Autónoma

La tarea de navegación se relaciona con la habilidad de un actor de establecer su ubicación haciendo uso de información del ambiente, y poner en uso esta información para el objetivo de desplazarse hacia una meta. En la navegación visual, este objetivo es realizado haciendo uso de imágenes del ambiente tanto para ubicarse como para construir el mapa del ambiente. El caso más común en esta situación viene siendo aquel en que el actor tiene una visión en primera persona del ambiente, equivalente a la forma en que vemos los humanos.

La navegación autónoma es un tema de alto interés en el área de robótica, en donde técnicas de Simultaneous Localization and Mapping (vSLAM) [14] en conjunto con técnicas de planificación de movimiento se encargan de cumplir los objetivos anteriormente mencionados. Sin embargo, no es el objetivo de este trabajo proponer nuevos algoritmos para la navegación, sino que la navegación es utilizada como una tarea para evaluar técnicas de aprendizaje reforzado. En este ámbito, la tarea de navegación visual se asemeja a un problema del mundo real y significan un problema desafiante debido a la alta complejidad visual presente en ambientes tridimensionales y el conocimiento parcial del estado del ambiente. Debido a esto muchos trabajos han utilizado tareas de navegación para evaluar nuevas arquitecturas y algoritmos de aprendizaje reforzado profundo, tanto de propósito general como específicas para la tarea de navegación.

En [7] se utilizan tareas auxiliares de predicción de profundidad en la imagen y detección de cierre de bucles en la trayectoria descrita, con el objetivo de reforzar el aprendizaje de características que serán de utilidad en la tarea principal. De modo más general, en [8] se utilizan tareas auxiliares relacionadas con maximizar el cambio en la intensidad de los pixeles, predicción de recompensa y maximización de la activación de las capas de la red. En [9] se alimenta el sistema de aprendizaje reforzado con una señal de recompensa intrínseca, que permite al sistema aprender en ausencia de un recompensa extrínseca. La recompensa intrínseca se obtiene a partir de la capacidad del actor de predecir su siguiente estado.

C. Características Visuales Complejas Calculadas por la Retina

La retina es una parte del sistema nervioso central que se ubica fuera del cerebro. Consiste en un conjunto de capas que contienen diferentes tipos de células y funciones. La diversidad de propiedades fisiológicas, circuitos y cómputo de información demuestran que no es un solo codificador de luz a electricidad, sino un capa de preprocesamiento, que se encarga de extraer señales relevantes del mundo visual que son críticas para la supervivencia de los animales [15], [16].

En la retina, la detección de características visuales surge como resultado de los circuitos subyacentes y la interacción entre los diferentes tipos de células. En general, cada uno de estos extractores de características tiene asociado con cierto tipo de células ganglionares de la retina (RGC por sus siglas en inglés), con cierta morfología y circuitería que proporciona su papel funcional. En primates, cada tipo de RGC parece ocupar todo el campo visual [17] sugiriendo cálculos similares. Sin embargo, un estudio reciente en primates mostró que el mismo tipo de RGC calcula diferentes propiedades del campo visual dependiendo de su ubicación dentro de la retina (central versus periférico) [18]. Más aún, un estudio reciente en un roedor diurno evidenció que las diferentes concentraciones de RGC en las regiones central y periférica está asociado a diferentes propiedades funcionales de lectura visual, procesando la información de manera diferente [4], y pudiendo aplicar dichas funcionalidades a la extracción de características de una escena visual.

III. Métodos Propuestos

A. Asynchronous Advantage Actor Critic (A3C)

Un algoritmo popular en aprendizaje reforzado profundo es el *Asynchronous Advantage Actor Critic (A3C)* [19]. Este algoritmo utiliza agentes en paralelo para generar experiencias descorrelacionandas, ofreciendo así mayor cobertura de exploración mientras utiliza los recursos computacionales de forma eficiente.

Este algoritmo es de tipo actor-critico (actor-critic) los cuales poseen dos componentes: el crítico que se encarga de mantener un estimativo de la función valor y el actor que mantiene una política, la cual es mejorada haciendo uso de las valoraciones del crítico. En el algoritmo A3C, el actor es

una red neuronal parametrizada por el conjunto de parámetros θ_π, que estiman la política a partir del estado (la entrada a la red). Esta es optimizada mediante un algoritmo de *gradiente descendente*, utilizando la función de pérdida

$$\mathcal{L}_\pi(\theta_\pi) = -\frac{1}{T}\sum_{t=0}^{T}\log\left(\pi(a_t|s_t,\theta_\pi)\right)A_t, \quad (1)$$

donde $\pi(a_t|s_t)$ corresponde a la verosimilitud de una acción dado un estado s_t y una política π [19]. $A_t = Q_t - V_t \approx (R_t + \gamma^T V_T) - V_t$, es el llamado valor de ventaja y que representa la "ventaja", en cuanto a recompensa descontada, que se obtuvo al elegir cierta acción por sobre la recompensa descontada esperada según la función valor V, en donde γ corresponde al factor de descuento utilizado para el cálculo de la recompensa descontada según el modelo de horizonte infinito, y R_t es la recompensa descontada.

El crítico es quien se encarga de mantener la aproximación de la función valor, que similar al actor hace uso de una red neuronal parametrizada con parámetros θ_V. La optimización de esta red es realizada mediante una función de pérdida de error cuadrático medio

$$\mathcal{L}_V(\theta_V) = \frac{1}{T}\sum_{t=0}^{T}(R_t - V_t(\theta_V))^2. \quad (2)$$

Adicionalmente se define un término de regularización de entropía $\mathcal{L}_H$. En la ecuación 3, $H(\pi(s_t|\theta_\pi))$ denota la entropía de la distribución de la política, con lo que el rol del término $\mathcal{L}_H$ es el de incentivar la exploración y evitar que la política converja a un proceso determinista, ya que al intentar maximizar la entropía hará que la política tienda a igualar la probabilidad de elección de todas las acciones, incluyendo aquellas acciones que no son consideradas las más óptimas por el sistema.

$$\mathcal{L}_H(\theta_\pi) = -\sum_{t=0}^{T} H(\pi_t(s_t|\theta_\pi)). \quad (3)$$

Finalmente, la función de pérdida total para la optimización dada por una suma ponderada de las pérdidas definidas es

$$\mathcal{L}_{A3C} = \mathcal{L}_\pi + \alpha\mathcal{L}_V + \beta\mathcal{L}_H. \quad (4)$$

En el proceso de entrenamiento cada uno de los agentes mantiene sus propios valores para los parámetros $(\theta' = \theta'_\pi \cup \theta'_V)$ de la red que se utilizan para la toma de decisiones durante la simulación y el cálculo de los gradientes. Los parámetros de cada agente son actualizados periódicamente, copiando los parámetros globales $(\theta = \theta_\pi \cup \theta_V)$ a medida que estos son actualizados. El proceso realizado por cada agente consiste en primer lugar en realizar una copia de los parámetros globales. Luego, utilizando la política resultante de estos parámetros, en conjunto con alguna estrategia de decisión (por ejemplo, ϵ-greedy), actuar sobre el ambiente por T pasos, recibiendo recompensas r_t. Posteriormente, se calcula el gradiente de la función de pérdida $d\theta$, utilizando también los parámetros propios. Finalmente se realiza la actualización de los parámetros globales, utilizando el gradiente calculado junto con alguna variedad de gradiente descendente, tras lo cual se reinicia el proceso el cual se repite por la cantidad de pasos que se desee.

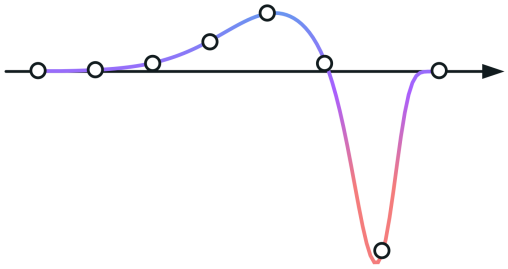
B. *Módulo Retinal Bio-Inspirado*

En la forma en que se aborda la tarea de navegación, la entrada del sistema (el estado), corresponde a una serie de imágenes que representa lo que el agente ve del mundo que le rodea. En el **primer modelo propuesto**, esta entrada visual es en primera instancia procesada por una etapa fija, compuesta por filtros bio-inspirados que siguen el comportamiento de campos receptivos encontrados en la retina de roedores diurnos [4].

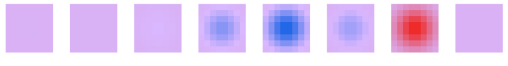
Estos filtros se caracterizan por tener una componente espacial y una temporal, separables, que actúan en secuencia, constituyendo así un mecanismo básico para la detección de movimiento en visión [20]. La componente espacial de los filtros es modelada como una función Gaussiana bidimensional de parámetro σ. La componente temporal modula a la componente espacial usando la siguiente función de activación:

$$f(t) = A_1\left(\frac{t}{\tau_1}\right)^n e^{-n(t/\tau_1-1)} - A_2\left(\frac{t}{\tau_2}\right)^n e^{-n(t/\tau_2-1)}, \quad (5)$$

en donde los parámetros A_1, A_2, τ_1, τ_2 y n definen la dinámica temporal del filtro (ver Figura 1). Estos parámetros fueron escogidos de manera de obtener una distribución similar a aquellos datos obtenidos en [4]. A partir de estos datos se extrajeron dos filtros: uno con un campo receptivo espacial amplio y respuesta temporal rápida, y otro con un campo receptivo espacial más angosto y respuesta temporal lenta, que emulan células de la retina ubicadas en la periferia y el centro, respectivamente. Los dos filtros anteriormente mencionados son aplicados independientemente sobre cada uno de los canales de la entrada resultando en 6 mapas de características, considerando una entrada con tres canales de color (en este caso RGB).



(a) Componente temporal



(b) Componente espacial

Fig. 1. Características de los filtros retinales. La Figura (a) muestra la componente temporal para uno de los filtros. Los puntos marcados a lo largo de la curva indican los puntos en que la función es muestreada al generar el filtro. La Figura (b) muestra la componente espacial modulada por la componente temporal en los puntos mostrados en la Figura (a).

Los filtros se implementaron como dos etapas de convolución. La primera correspondiente a la característica espacial que corresponde a una convolución utilizando un kernel de $1 \times 8 \times 8 \times 1$ pixeles. Esto es operando sobre regiones de 8×8 pixeles sobre un cuadro a la vez y sobre un único canal. Esta convolución es realizada utilizando un paso de 4

pixeles. Luego, para la componente temporal, se aplica una nueva convolución sobre los resultados utilizando un kernel de tamaño $8 \times 1 \times 1 \times 1$. Esto es operando sobre una única posición en la imagen, pero tomando los últimos 8 cuadros. En la Figura 2 se muestra un esquema de la operación del módulo retinal que incorpora estos filtros.

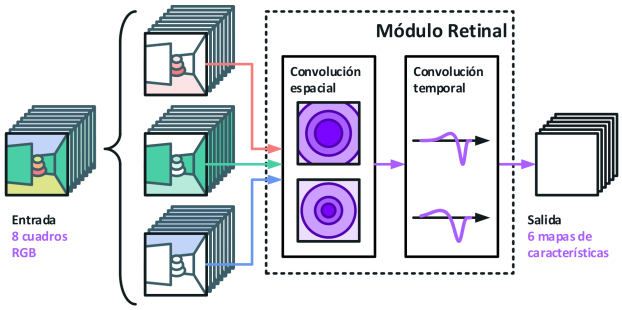


Fig. 2. Esquema del módulo retinal. A la serie de imágenes de entrada se les aplican dos operaciones de convolución. La primera aplica dos kernels Gaussianos de diferente tamaño operando en cada cuadro y cada canal de color por separado. Esta representa la componente espacial del filtro retinal. La segunda convolución aplica la componente temporal de los filtros y opera sobre todos los cuadros a la vez.

1) Arquitectura de Red Neuronal: El resultado del procesamiento realizado por el módulo retinal es alimentada a una serie de capas de redes neuronales, consistentes en: una capa convolucional, una capa completamente conectada, una capa recurrente y las capas de salida correspondientes a un último par de capas completamente conectadas. Estas capas se muestran en el esquema del modelo mostrado en la Figura 3 (a).

La capa convolucional consiste en 32 kernels de 4×4 pixeles operando con un paso de 2 pixeles. La salida de esta capa entra en una red completamente conectada de 256 unidades. Estas primeras dos capas, la capa convolucional y la capa completamente conectada usan como función de activación la función ELU. El resultado de la capa completamente conectada entra en una red recurrente LSTM de 256 unidades que finalmente mapea a la política π y a la predicción de la función valor V , mediante las capas de salida. Al vector de salida correspondiente a la política se les aplica la función de activación SoftMax para obtener una distribución de probabilidades sobre las acciones.

C. Tareas Auxiliares de Predicción

Con el **segundo modelo propuesto** se busca atacar el problema del aprendizaje en ausencia de recompensa incorporando la idea de tareas auxiliares. Las tareas auxiliares propuestas toman su inspiración del concepto de motivación intrínseca [6]. Una forma de modelar un sistema de motivación intrínseca es mediante la novedad predictiva según el cual las situaciones interesantes serán aquellas que sean difíciles de predecir. En el modelo propuesto, se utilizará la idea de aprender de aquellas situaciones que no se pueden predecir, mediante el uso de las tareas auxiliares.

El modelo que se utilizará para evaluar el desempeño de estas tareas auxiliares es similar al modelo presentado anteriormente para el módulo retinal como se muestra en la Figura 3 (b), con la diferencia de que éste es reemplazado por una capa convolucional y únicamente se utiliza la última observación como entrada. Notar que en este modelo, tras pasar las capas convolucionales y la siguiente capa feed-forward se tiene un resultado que depende solo de la observación del estado, por lo que es posible considerar este valor como una representación aprendida del estado. Esta representación aprendida además de utilizarse para generar la política y la estimación valor, durante el entrenamiento se utiliza como entrada a un par de capas complemente conectadas. La primera de estas capas toma como entrada, además de la representación aprendida, la acción realizada y su salida se utiliza para estimar el valor que tomará la representación interna en el siguiente paso, es decir, tras haber realizado la acción. La segunda de las capas completamente conectada, tiene como entrada el valor que tomará la representación aprendida tras realizar la siguiente acción. La salida de esta capa se utiliza para estimar cuál fue la acción que se realizó para pasar de una representación a la otra.

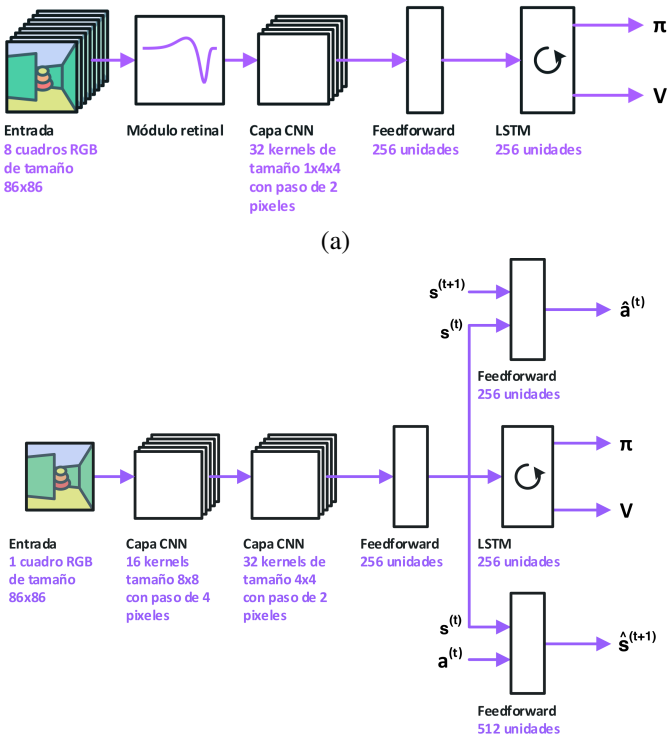


Fig. 3. Esquemas de los mecanismos propuestos. En (a) la entrada visual entra al módulo retinal. La salida de éste pasa a través de una capa convolucional una capa completamente conectada y una capa recurrente para finalmente llegar a una capa de salida que resulta en la política y la aproximación de la función valor. En (b) solo la última entrada visual entra directamente a las capas convolucionales. Además, la salida de la capa completamente conectada es utilizada para predecir el próximo estado y la próxima acción utilizando capas completamente conectadas adicionales.

La primera tarea auxiliar, que se le llamará **predicción de estado**, empuja a la red a ser capaz de predecir la siguiente representación del estado, con esto se buscan que la red aprenda sólo aquellas características del ambiente que sean predecibles. Por otro lado, la **predicción de acciones**

busca que las características aprendidas sean relevantes con el método en que el agente actúa con el ambiente, para evitar aprender características predecibles pero irrelevantes. Para realizar el entrenamiento de este modelo se agregan a la pérdida del algoritmo A3C (ecuación 4) dos términos asociados a las predicciones: $\mathcal{L} = \mathcal{L}_{A3C} + \delta\mathcal{L}_s + \epsilon\mathcal{L}_a$. En donde, $\mathcal{L}_s$ y $\mathcal{L}_a$ son las pérdidas asociadas a la predicción de la representación interna del estado y a la acción, respectivamente, cuya importancia en el proceso de entrenamiento es controlado por los parámetros δ y ϵ.

La pérdida $\mathcal{L}_s$ es el error cuadrático medio entre la predicción del estado $\hat{s}_{t+1}$ realizado por la red y el valor s_{t+1} que efectivamente toma el estado en el paso siguiente. La pérdida $\mathcal{L}_a$ es la entropía cruzada entre la predicción $\hat{a}_t$ y la acción que efectivamente se realizó para la transición.

IV. Experimentos

A. Ambiente

La generación de experiencias para el proceso de aprendizaje, para ambos métodos propuestos, se realizó utilizando el ambiente de simulación DeepMind Lab [21]. Este simulador provee un ambiente tridimensional simple y simulación de física básica, y busca proveer tareas complejas en ambientes visuales diversos. Gracias a su simplicidad es posible ejecutar múltiples instancias a la vez en una misma máquina. El simulador es operado mediante una API de Python con la que se pueden rescatar directamente las entradas visuales y recompensas, para ser alimentadas a un algoritmo de aprendizaje reforzado.

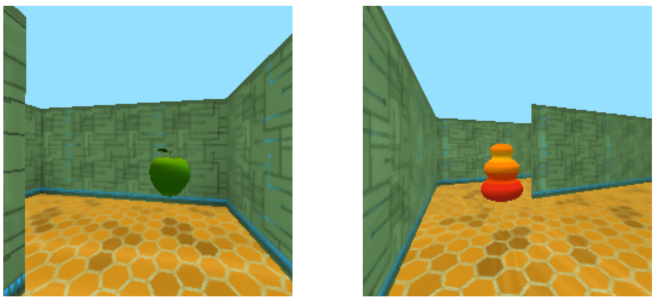


Fig. 4. Capturas del ambiente *DeepMind Lab* para tareas de navegación. La captura de la izquierda muestra una de las manzanas que entrega una recompensa al agente, mientras que en la derecha se muestra el objeto que representa la meta a alcanzar por el agente.

Este ambiente provee tareas de navegación en donde un agente con visión egocéntrica del mundo debe moverse a través de un laberinto en búsqueda de una meta. En estas tareas el agente es recompensado al recolectar manzanas que se encuentran dispersas en el laberinto, las que entregan una recompensa de 1 punto, o al llegar a la meta, que entrega 10 puntos (ambos objetivos se muestran en la Figura 4). El encontrar la meta causa además que el mapa se reinicia, volviendo a hacer aparecer las manzanas recolectadas y moviendo al agente a una nueva posición. Un episodio contempla una cantidad fija de tiempo dentro del cual el simulador ejecuta un número fijo de pasos de simulación (o pasos de ambiente), que corresponden a los pasos. En las tareas de navegación, el agente puede alcanzar múltiples veces la meta en un mismo episodio, lo que será necesario para maximizar la recompensa total.

B. Modelos a Evaluar

Para la evaluación del modelo de filtros retinales se realizaron experimentos utilizando los siguientes modelos:

Filtros Retinales: El procesamiento visual se realiza utilizando el módulo retinal propuesto, empleando dos filtros fijos de $8\times8\times8\times1$ pixeles (i.e. considerando 8 cuadros, regiones de 8×8 pixeles y 1 canal de color) actuando independientemente sobre cada uno de los canales de entrada. En la Figura 3 se muestra este modelo.

Filtros Temporales: Similar a el modelo de filtros retinales, pero utiliza 6 filtros de $8\times8\times8\times3$ con pesos entrenados en lugar de fijos.

Modelo base: El procesamiento visual es realizado utilizando una red convolucional con 16 filtros de tamaño $1\times8\times8\times3$, que a diferencia de los otros modelos, no posee una componente temporal y utiliza únicamente el cuadro más reciente como entrada. Este es el modelo LSTM A3C usado en [7].

Para la evaluación del modelo de tareas auxiliares, solamente se utilizó el Modelo base con la arquitectura mostrada en Figura 3(b).

C. Entrenamiento

En cada experimento, el entrenamiento se realizó durante 10^8 pasos de ambiente equivalentes a $2.5\cdot10^7$ pasos del agente (es decir, la cantidad de acciones realizadas). La diferencia entre la cantidad de pasos del agente y pasos del ambiente se debe a que una vez que una acción es escogida esta se repite durante 4 pasos de ambiente. Las acción a realizar se escoge aleatoriamente según la distribución dada por la salida π_t de la red.

Ambos métodos propuestos son entrenados utlizando el algoritmo A3C. Para la generación de experiencias se utilizan 16 agentes en paralelo, cada uno con su propio ambiente de simulación. Como parámetros para la función de pérdida se utilizó un valor de 0.5 para α y valores en el intervalo $[10^{-4}, 10^{-3}]$ para β. Como factor de descuento se utilizó el valor 0.99. Se utilizó un horizonte de 50 pasos del agente para la actualización de los parámetros globales en el algoritmo A3C y para el desenrollamiento de las capas recurrentes LSTM. La optimización de los parámetros de la red se realiza con el algoritmo RMSProp [22], un algoritmo de gradiente descendente estocástico. Para este algoritmo se utilizó una taza de aprendizaje de 10^{-5} y un factor de momentum de 0.95[1].

Una vez finalizados los procesos de entrenamiento, se utilizan los valores finales de los parámetros para simular 100 episodios de la tarea para la que fueron entrenados. El promedio de las recompensas que se obtienen en estos episodios se utilizas como métrica de desempeño para la comparación de los resultados.

[1]El código de la implementación acá utlizada se encuentra disponible en el repositorio https://github.com/HansLehnert/rl

V. Resultados

A. Modelo Retinal Bio-Inspirado

Para cada uno de los modelos se realizaron experimentos para tres valores diferentes del parámetro β, que controla la influencia de la regularización de entropía, debido a la gran variabilidad de los resultados que se observó asociada a este parámetro durante la realización de los experimentos.

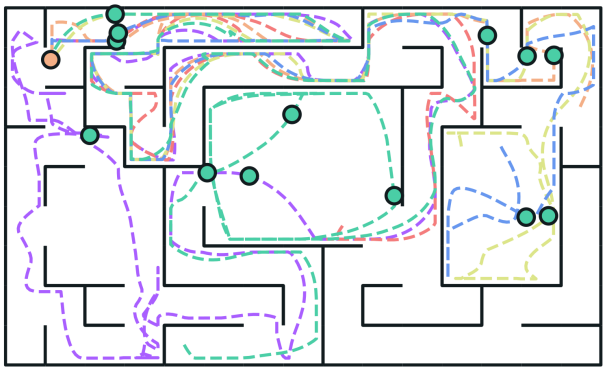
Una vez terminado el entrenamiento, se utilizaron los modelos entrenados para simular 100 episodios de la tarea de navegación, para el laberinto estático y el laberinto aleatorio, para tres valores distintos del parámetro β. La Tabla I muestra los resultados obtenidos para cada uno de los datos analizados. Todos los modelos lograron alcanzar un desempeño satisfactorio para alguno de los valores del parámetro. En particular, el modelo retinal propuesto logró un desempeño favorable en comparación con los otros modelos para el caso del laberinto estático. Los filtros temporales y filtros retinales alcanzan desempeños comparables con el caso de filtros simples a pesar de utilizar una menor cantidad de características de entrada.

TABLA I
Recompensa promedio obtenida por los modelos entrenados para cada uno diferentes valores de influencia para los modelos de *filtros retinales* (FR), *filtros temporales* (FT) y *Modelo Base* (MB), utilizando el laberinto estático y el aleatorio

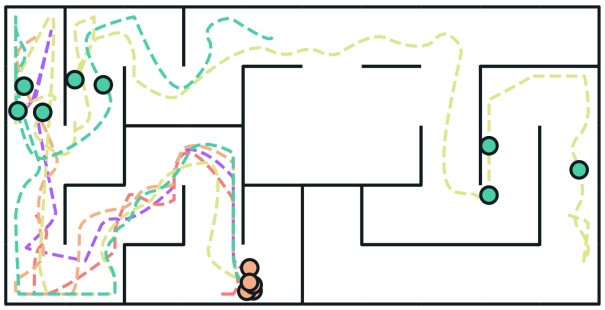
Modelo	Recompensa promedio					
	L. Estático			L. Aleatorio		
β	1e-4	5e-4	1e-3	1e-4	5e-4	1e-3
FR	19.65	36.20	34.99	18.13	45.88	4.54
FT	6.75	41.49	4.78	47.52	12.99	50.02
MB	5.22	5.41	5.00	50.39	42.91	41.31

La Figura 5 muestra algunas trayectorias realizadas por los agentes. Para el laberinto estático, en general se pudo observar que aquellos agentes que alcanzaron un alto desempeño (evaluación final mayor a 30) obtuvieron la capacidad de ubicarse y navegar correctamente y aprender la ruta hasta la meta. Para estos agentes, se pudo observar que las diferencias de desempeño final se traducen en la facilidad con que se desplazan. Aquellos con menor puntaje exhiben un movimiento más "tosco", por ejemplo, se atascan con paredes que sólo ven parcialmente o dan giros innecesarios. Para el caso del laberinto aleatorio, en los agentes que logran una mayor desempeño final también se observa la capacidad de localizarse y aprender una trayectoria a través del laberinto. Sin embargo, para esta tarea aprender una única ruta no es el comportamiento óptimo, debido a que las recompensas cambian de lugar entre episodios. Esto significó que estos agentes obtuvieron una alta recompensa cuando la meta se encontraba en la ruta que aprendieron, pero la capacidad de buscar y recordar dónde se encuentra la meta no fue adquirida.

Las trazas de entrenamiento del sistema mostradas en [5] muestran cierta inestabilidad en el proceso de aprendizaje. Es por esto, que se evalúa aplicar diferentes tamaños de batch y analizar su efecto en la recompensa acumulada. En los experimentos mostrados en [5] se utiliza un tamaño de batch igual a 1, lo que hace al sistema sensible a singularidades de alguno de sus agentes. Se propone evaluar con dos tamaños



(a) Laberinto aleatorio



(b) Laberinto estático

Fig. 5. Muestra de las trayectorias realizadas durante un episodio por los agentes entrenados con el modelo de filtros retinales. Los círculos verdes muestran la posición en que el agente recolecta una manzana y los círculos naranja el lugar en donde alcanza la meta. Cada vez que el agente alcanza la meta comienza un nuevo recorrido, denotado por los trazos de diferente color.

de batches, 1 y 32, para los tres valores del parámetro de regularización β. Los resultados se muestran en la Figura 6, en donde se evidencia claramente un mejor desempeño para el tamaño del batch igual a 1. Tamaños grandes generan un menor número de actualizaciones en los pesos, ya que, los valores de los gradientes son promediados logrando pequeños avances de la solución.

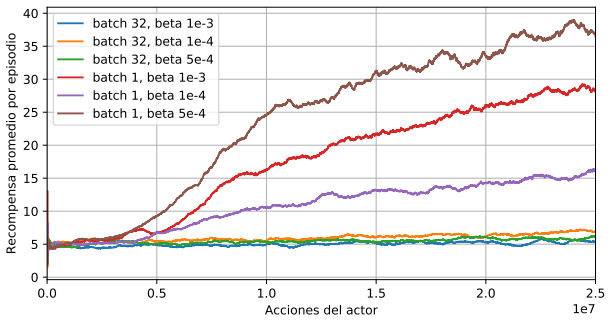


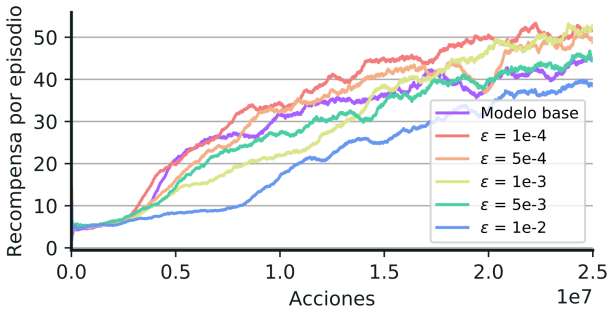
Fig. 6. Desempeño del sistema para dos tamaños de batch. Se compara el desempeño del modelo de Filtro Simple utilizando tamaños de batch 1 y 32.

B. Tareas Auxiliares de Predicción

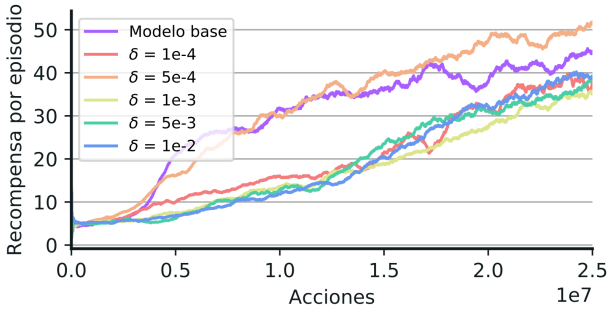
Para el caso de las tareas auxiliares, se probaron actuando tanto independientemente como actuando en conjunto. Los modelos con tareas auxiliares se entrenaron únicamente en el

laberinto aleatorio y usando un valor para β de 10^{-3} para el modelo de Filtro Temporal.

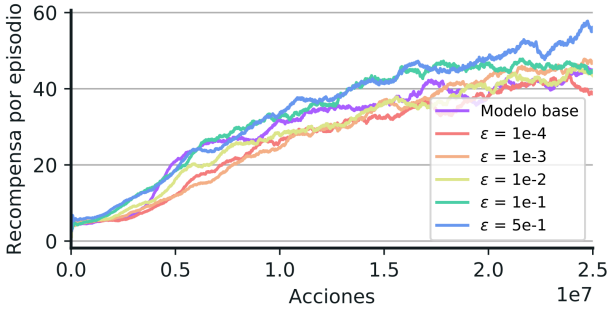
En la Figura 7 se muestra la progresión del desempeño para cada uno de los casos evaluados. A diferencia de los resultados mostrados en la Sección V-A, las tareas auxiliares no mostraron la introducción de inestabilidad al proceso.



(a) Predicción de acción



(b) Predicción de estado



(c) Predicción de estado y acción

Fig. 7. Curvas de aprendizaje para modelos con tareas auxiliares.

Para el caso de la **predicción de la acción**, es decir, variando el parámetro ϵ, se obtienen resultados mejores que el Modelo Base. Lo mismo se observa utilizando ambas tareas combinadas, donde el experimento con $\{\delta, \epsilon\} = \{5 \cdot 10^{-4}, 5 \cdot 10^{-1}\}$ muestra un resultado considerablemente mayor que el caso base. Diferente es el caso de la **predicción de estado**, en donde esta tarea por sí sola no resultó favorable en cuanto a los resultados finales. Esto se puede deber a que esta tarea por si sola favorece que se aprendan aquellas características que serían fáciles de predecir, pero no necesariamente relevantes para la tarea. En la tarea abordada, por ejemplo, el cielo es siempre azul y se encuentra en la misma región del campo visual, por lo que sería fácil predecir que en el futuro seguirá siendo del mismo color y seguirá en el mismo lugar. Para entrenar los modelos se utilizaron 16 agentes del algorítmo A3C en una máquina con GPU Nvidia Tesla P100, CPU Intel Xeon E5-2630 v4 con 40 núcleos. El entrenamiento[2] tomó alrededor de 12 horas, para cada uno de los casos. Una evaluación del modelo entrenado toma entre 3 a 4 ms.

VI. Discusión y Conclusiones

En este trabajo se presentaron dos modelos que apuntan a mejorar el desempeño de técnicas de aprendizaje reforzado mediante el uso de mecanismos bio-inspirados. El primero utiliza una etapa de pre-procesamiento basado en modelos retinales para facilitar la extracción características de interés de la entrada visual. El segundo utiliza tareas auxiliares inspiradas en sistemas de motivación intrínseca. Para el modelo que incorporó los filtros retinales, se lograron resultados comparables a aquellos más complejos y que extraen una mayor cantidad de características. Notablemente esto fue realizado con filtros cuya característica fue construida a partir de datos biológicos. Sin embargo, se debe mencionar que existen problemas en cuanto a la estabilidad del proceso de entrenamiento que se mejoran con el segundo modelo propuesto de esta contribución, el de tareas auxiliares de predicción.

Los experimentos en el laberinto estático son prometedores, mostrando una clara ventaja en la rapidez del aprendizaje para el caso de los *Filtros Retinales*. Sin embargo, en los experimentos realizados en el laberinto aleatorio no se logró el comportamiento esperado, en que el agente explora en búsqueda de la meta y recuerda su posición. En lugar de esto el comportamiento obtenido se centra en aprender una única ruta principal a través del laberinto. Esta falencia puede deberse a que el simulador presenta un entorno poco variado en relación a un escenario real, lo que queda como perspectiva a futuro, utilizando quizás, algún agente robótico. Otra perspectiva interesante sería colocar el agente en un laberinto sin disposición fija, de manera que el éste no pueda valerse de una ruta aprendida.

En cuanto a las tareas auxiliares, se observó que usando sólo la tarea de predicción de estado se obtiene una disminución del desempeño en la mayoría de los experimentos. Por otra parte, al usar únicamente la tarea de predicción de acción vemos una ligera mejora en los resultados, que se ve potenciada al utilizarse ambas tareas en conjunto. Además se observó un rango relativamente amplio de robustez en cuanto a los parámetros de influencia de las tareas, permitiendo obtener consistentemente resultados de entrenamiento satisfactorios, lo que apuntaría a que su inclusión en un modelo no debería conllevar resultados negativos mejorando la estabilidad del sistema de aprendizaje.

Pese a que no fue posible realizar un prueba extensiva de los modelos presentados, principalmente debido a los largos tiempos requeridos para los procesos de entrenamiento, los experimentos realizados entregan indicios positivos acerca de la efectividad de los mecanismos propuestos. En particular, la arquitectura de *Filtros Retinales* muestra que es factible el uso de modelos biológicos para simplificar un modelo de redes neuronales y aún así lograr resultados satisfactorios, abriendo una nueva e interesante veta de exploración que une la neurociencia con la inteligencia artificial.

[2]Más detalles de la implementación pueden encontrarse en https://repositorio.usm.cl/handle/11673/46319

AGRADECIMIENTOS

Financial support: CONICYT-Basal Project FB0008 and FB0821, AFOSR Grant Nro. FA9550-19-1-0002, CNRS-PICS Nro. 07844.

REFERENCIAS

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Aug. 2017.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. v. d. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[4] M.-J. Escobar, C. Reyes, R. Herzog, J. Araya, M. Otero, C. Ibaceta, and A. G. Palacios, "Characterization of Retinal Functionality at Different Eccentricities in a Diurnal Rodent," *Frontiers in Cellular Neuroscience*, vol. 12, p. 444, 2018.

[5] H. Lehnert, M.-J. Escobar, and M. Araya, "Retina-inspired visual module for robot navigation in complex environments," in *To appear in 2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[6] P.-Y. Oudeyer and F. Kaplan, "What is Intrinsic Motivation? A Typology of Computational Approaches," *Frontiers in Neurorobotics*, vol. 1, Nov. 2007.

[7] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to Navigate in Complex Environments," *CoRR*, 2016.

[8] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement Learning with Unsupervised Auxiliary Tasks," *CoRR*, 2016.

[9] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven Exploration by Self-supervised Prediction," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 2778–2787. [Online]. Available: https://arxiv.org/abs/1705.05363

[10] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, and D. Kumaran, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, pp. 429–433, 2018.

[11] G. Tesauro, "Temporal difference learning and TD-Gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[12] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution Strategies as a Scalable Alternative to Reinforcement Learning," *arXiv:1703.03864 [cs, stat]*, Mar. 2017, arXiv: 1703.03864. [Online]. Available: http://arxiv.org/abs/1703.03864

[13] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu, "IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures," *arXiv:1802.01561 [cs]*, Feb. 2018, arXiv: 1802.01561. [Online]. Available: http://arxiv.org/abs/1802.01561

[14] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, Jun. 2017.

[15] T. Gollisch and M. Meister, "Rapid Neural Coding in the Retina with Relative Spike Latencies," *Science*, vol. 319, no. 5866, pp. 1108–1111, Feb. 2008.

[16] ——, "Eye Smarter than Scientists Believed: Neural Computations in Circuits of the Retina," *Neuron*, vol. 65, no. 2, pp. 150–164, Jan. 2010.

[17] J. L. Gauthier, G. D. Field, A. Sher, M. Greschner, J. Shlens, A. M. Litke, and E. J. Chichilnisky, "Receptive Fields in Primate Retina Are Coordinated to Sample Visual Space More Uniformly," *PLoS Biology*, vol. 7, no. 4, pp. e1 000 063–9, Apr. 2009.

[18] R. Sinha, M. Hoon, J. Baudin, H. Okawa, R. O. L. Wong, and F. Rieke, "Cellular and Circuit Mechanisms Shaping the Perceptual Properties of the Primate Fovea," *Cell*, vol. 168, no. 3, pp. 413–426.e12, Jan. 2017.

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, 2016, pp. 1928–1937.

[20] A. Borst and M. Egelhaaf, "Principles of visual motion detection," *Trends in Neurosciences*, vol. 12, no. 8, pp. 297–306, Aug. 1989.

[21] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, J. Schrittwieser, K. Anderson, S. York, M. Cant, A. Cain, A. Bolton, S. Gaffney, H. King, D. Hassabis, S. Legg, and S. Petersen, "DeepMind Lab," *arXiv:1612.03801 [cs]*, Dec. 2016, arXiv: 1612.03801. [Online]. Available: http://arxiv.org/abs/1612.03801

[22] G. Hinton, N. Srivastava, and K. Swersky, *Lecture 6a Overview of mini-batch gradient descent*.

Hans Lehnert es graduado del programa de Magister en Ciencias de la Ingeniería Electrónica de la Universidad Técnica Federico Santa María (UTFSM). Sus prinicipales intereses son el aprendizaje reforzado y arquitecturas de computación paralelas.

Mauricio Araya es profesor auxiliar del Departamento de Electrónica de la Universidad Técnica Federico Santa María (UTFSM), Valparaíso, Chile, e Investigador del Centro Avanzado de Ingeniería Eléctrica y Electrónica (AC3E), del Centro Científico Tecnológico de Valparaíso (CCTVal), y del Observatorio Virtual Chileno (ChiVO). Sus principales intereses son el aprendizaje reforzado, la astroinformática, la teória de la información y la ciencia de datos aplicada a ciencias.

Rodrigo Carrasco-Davis es asistente de investigación en el departamento de Electrónica de la Universidad Técnica Federico Santa María (UTFSM), Valparaíso, Chile, y asistente de investigación en el Departamento de Ingeniería Eléctrica de la Universidad de Chile, Santiago, Chile. Sus principales intereses de investigación son las aplicaciones de aprendizaje de máquinas en astronomía, y la neurociencia computacional.

Maria-José Escobar es profesora adjunta del Departamento de Electrónica de la Universidad Técnica Federico Santa María (UTFSM), Valparaíso, Chile, e Investigadora Principal del Centro Avanzado de Ingeniería Eléctrica y Electrónica (AC3E). Sus principales intereses de investigación son la visión biológica, neurociencia computacional, inteligencia artificial y la robótica cognitiva. En el AC3E, lidera la línea de investigación de Análisis de Datos e Inteligencia Artificial.