



Neuroevolución Profunda: Aplicaciones en Ciudades Inteligentes

Andrés Camero*, Enrique Alba†

Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación

*Doctorando, andrescamero@uma.es

†Director, eat@lcc.uma.es

Resumen—En la última década la aparición del Deep Learning y las redes neuronales profundas ha revolucionado la sociedad. Cada vez son más las aplicaciones que hacen uso de ella, desde coches autónomos, hasta asistentes robotizados. Sin embargo, a medida que estas redes crecen, nos enfrentamos a nuevos desafíos. Un problema particular es la búsqueda de una configuración o hiperparámetro óptima, es decir encontrar la red que mejor se adapta a un problema dado. Se han propuesto diversos enfoques para abordar la hiperparametrización, pero en la práctica pocos han sido adoptados por el excesivo tiempo de cómputo que requieren. No obstante, en los últimos años ha surgido un nuevo acercamiento: la neuroevolución profunda. Basándose en el uso de metaheurísticas avanzadas, ésta ofrece una nueva forma de solucionar el problema, la cual está permitiendo encontrar soluciones sin precedentes y en tiempos cada vez más razonables, abriendo el camino a su adopción práctica. En este trabajo presentamos un proyecto de investigación para abordar el problema de la optimización de la hiperparametrización de una red neuronal recurrente, un tipo de red profunda especialmente buena para resolver problemas que requieran aprender dependencias temporales. Concretamente, proponemos una solución basada en la neuroevolución profunda, la cual validaremos usando tres problemas en el contexto de la ciudad inteligente.

Index Terms—Deep neuroevolution, Deep learning, Hyperparameter optimization, Recurrent neural network, Smart City

I. INTRODUCCIÓN

En las últimas décadas, gracias al vertiginoso aumento de la potencia de cómputo y almacenamiento de datos, se ha evidenciado un resurgimiento del interés por desarrollar redes neuronales artificiales (ANN). Las “antiguas” técnicas, combinadas con esta nueva potencia computacional han permitido obtener resultados sorprendentes: desde coches autónomos, hasta asistentes personales robotizados [1].

En gran medida este resurgimiento ha sido impulsado por el *Deep Learning* (DL) [1] (aprendizaje profundo), que en términos simples consiste en diseñar y entrenar ANN (“lo antiguo”) de mayor complejidad y tamaño, con una cantidad de datos inmensa (“lo nuevo”). Sin embargo, a medida que estas redes crecen se vuelven más complejas, lo que plantea un nuevo desafío: ¿Cómo diseñamos una ANN profunda? Tradicionalmente el diseño de las ANN ha sido tarea de un experto, quien basado en su conocimiento, en técnicas y herramientas específicas, y en un arduo trabajo de prueba y error diseña una ANN que se adapta de buena forma al problema. Sin embargo, debido a la elevada complejidad de las ANN profundas (¡millones de parámetros!), esta forma de diseñar se torna ineficiente. La situación empeora conforme la cantidad de datos usados para entrenar la red crece, luego sólo el entrenamiento de una ANN profunda puede tardar días, por lo que un diseño óptimo puede tardar meses [2].

Dada la especial importancia que tiene encontrar un diseño óptimo (o hiperparametrización de la ANN) en el desempeño global, incluyendo tanto el resultado como el proceso de aprendizaje, se han realizado numerosas propuestas para automatizar la búsqueda de una solución [2]. Sin embargo, una amplia mayoría de las propuestas basa su funcionamiento en la prueba y el error, es decir se escoge un diseño, se entrena la red y se evalúa su resultado. Si bien este enfoque es capaz de entregar buenas soluciones, es muy costoso, por lo que en la práctica es poco utilizado. Una solución intuitiva para este tipo de técnicas consiste en limitar el esfuerzo del entrenamiento (i.e. terminar el entrenamiento de manera precoz), lo cual permite reducir significativamente el tiempo. No obstante, dada la elevada complejidad de los diseños de las ANN profundas, es posible que un buen diseño sea descartado debido a que es de lento aprendizaje.

Como solución alternativa al problema de la hiperparametrización surge la *Deep Neuroevolution* [3]–[5] (DN) (neuroevolución profunda), un conjunto de técnicas que permiten evolucionar el diseño de una ANN profunda, es decir la adaptación de la “antigua” neuroevolución [6] a los desafíos del DL. Utilizando metaheurísticas (e.g algoritmos genéticos, colonias de hormigas, recocido simulado, etc.) es posible navegar a través del espacio de diseños, encontrando así uno que se adapta de mejor manera al problema. Si bien a primera vista este también es un enfoque de prueba y error, existe una gran diferencia: la DN es una búsqueda dirigida que explota las características topológicas del problema y no se basa (necesariamente) en el entrenamiento de la red. Por lo tanto, la DN abre un camino que se espera permita mejorar los resultados actuales.

En este trabajo presentamos nuestro acercamiento a la DN, mostrando a través de aplicaciones reales en el dominio de la *Smart City* (SC) (ciudad inteligente) su desempeño en contraste con el estado del arte de la hiperparametrización. Concretamente, presentamos un plan de trabajo para el desarrollo de una tesis doctoral que busca contribuir en la hiperparametrización de una red neuronal recurrente (RNN), un tipo de ANN profunda que incluye conexiones neuronales que retroalimentan a la misma red, mediante técnicas de DN.

El resto de este trabajo se organiza de la siguiente forma: la siguiente sección esboza el estado del arte, la Sección III presenta nuestra propuesta, la Sección IV introduce el plan de trabajo, la Sección V presenta una discusión sobre la relevancia de nuestra propuesta y en la Sección VI presentamos las conclusiones preliminares del avance del trabajo.

II. ESTADO DEL ARTE

En esta sección presentamos brevemente el contexto actual de las ANN profundas. Primero revisamos su historia, del pasado al presente, luego nos centramos en el DL, para finalmente introducir el estado del arte en la hiperparametrización de ANN profundas.

II-A. Redes Neuronales Artificiales

El cerebro humano es una “máquina” capaz de realizar tareas muy complejas, como el reconocimiento de patrones, el control motriz de una extremidad o la percepción de estímulos sensitivos, de forma mucho más veloz que cualquier máquina inventada por el hombre [7]. Por este motivo se han (y siguen haciendo) realizado grandes esfuerzos para comprender su funcionamiento [8], [9].

Esta “máquina” está formada por cerca de 100 billones de neuronas (células del sistema nervioso especializadas en la recepción y conducción de estímulos), las cuales se comunican entre sí, formando complejos circuitos, los cuales son capaces de llevar a cabo la función cerebral [9]. Este modelo biológico fue la inspiración para que en 1943 McCulloch y Pitts [10] propusieran un nuevo modelo computacional: la ANN.

Una ANN es una red de unidades de cómputo (neuronas) comunicadas (o conectadas) entre sí, donde cada unidad realiza un cálculo a partir de una entrada y comunica su resultado (salida) a las unidades conectadas. Típicamente las conexiones reciben un peso, el cual indica que tan fuerte es la conexión entre dos unidades. Asimismo, las unidades de cómputo suelen agruparse en módulos o capas. Las ANN presentan propiedades y capacidades que son muy útiles para la resolución de problemas complejos, destacándose su no linealidad (*nonlinearity*), adaptatividad, tolerancia a fallos, entre otros [7]. Gracias a estas propiedades se ha logrado superar los resultados obtenidos en múltiples problemas.

Las ANN despertaron un gran interés y se desarrollaron rápidamente, sin embargo las primeras décadas de su evolución se vieron marcadas por técnicas que requerían de un gran esfuerzo para diseñar la ANN y de un amplio conocimiento del dominio del problema para transformarlo en algo que la ANN pudiese procesar [1], [11]. No fue sino hasta mediados de la década de 1980, cuando tomando como base el método *stochastic gradient descent*, se logró plantear (y comprender) como reducir la intervención humana: el método *backpropagation* (BP) [12], [13].

BP es un procedimiento para computar el gradiente de una función objetivo (*loss function*) con respecto a los pesos de una ANN [13]. Este método en una aplicación práctica de la *regla de la cadena*, cuyo eje central es que el gradiente de una función objetivo con respecto a la entrada de una capa (de neuronas) puede ser calculado de atrás hacia adelante a partir del gradiente de la salida de la misma capa. Esto permite propagar desde la salida hacia la entrada el error, ajustando así los pesos de la red.

Luego de más de una década de estudio (a fines de 1990), el método *backpropagation* y las ANN en general fueron dejadas de lado, debido a que se pensó que en la práctica no era posible

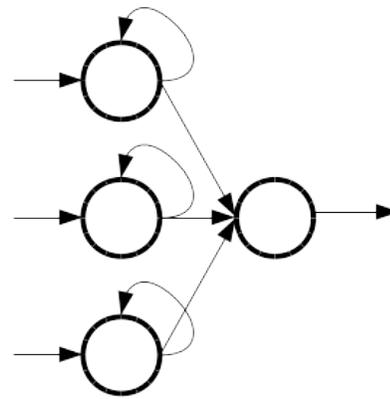


Figura 1. Red neuronal recurrente.

aprender conceptos complejos a partir de escasa información previa [1], o en otras palabras no se sabía cómo entrenar una ANN de múltiples capas ocultas que fuese capaz de modelar funciones no lineales altamente variables [14]. Sin embargo, casi una década después (a mediados de la década de 2000), gracias a la contribución de Hinton et al. [15], el interés por las ANN resurgió de la mano de un nuevo concepto: el DL.

II-B. Deep Learning

DL es un tipo de aprendizaje de máquina que permite entrenar ANN compuestas de múltiples capas ocultas, conocidas como ANN profundas (*deep neural network*, DNN) [1]. La teoría de circuitos sugiere que mientras más profunda sea la arquitectura de una ANN, más eficiente será en términos del número de elementos computacionales requeridos para representar funciones altamente no lineales y variantes [14].

Existen múltiples tipos de DNN: *feedforward* (FNN) [16], *radial-basis* [17], *recurrent* (RNN) [7], *modular* [17], [18], entre otros [7]. Particularmente, las RNN se caracterizan por propagar la información hacia adelante (al igual que las FNN) y por incluir al menos una conexión hacia atrás (retroalimentación *feedback*). Este *feedback* puede ser de una neurona sobre sí misma (*self-feedback*) o sobre otras neuronas. Asimismo, puede originarse desde las capas ocultas o desde la salida de la RNN. Por su arquitectura las RNN son en esencia las DNN de mayor profundidad.

La Figura 1 muestra una RNN. Gracias a las conexiones hacia atrás, las RNN son capaces de presentar un comportamiento dinámico no lineal [7].

A medida que la DNN se vuelve más compleja el número de parámetros crece. En términos simples, más neuronas implica más pesos, a más capas ocultas mayor no linealidad (más dificultad al propagar un error, e.g. BP), y en el caso de las RNN, la retroalimentación agrega más pesos y profundidad, es decir empeora aún más la situación. Luego, considerando la importancia que tiene la selección de un diseño apropiado de la DNN para la obtención de buenos resultados [2], se hace mandatorio abordar la hiperparametrización de manera inteligente.



II-C. Hiperparametrización

El problema de hiperparametrizar u optimizar una ANN consiste en encontrar una estructura de red, que incluye la selección del tipo de neuronas apropiadas (i.e. función de activación, memoria, etc.), el número de neuronas y su disposición (número de capas y conexiones), y un conjunto de pesos que permitan minimizar una función objetivo [2].

Existen múltiples algoritmos, técnicas y procedimientos que permiten abordar dicho problema, así como también diferentes enfoques, que consideran una de las *dimensiones* (estructura o pesos) o ambas a la vez [7].

Los métodos más populares en este contexto son BP [13] y *conjugate gradient* [19] (y sus variantes [2]). Ambos métodos permiten optimizar los pesos de una ANN con muy buenos resultados, sin embargo en problemas de alta dimensionalidad, es decir en el caso de DNN, los dos métodos tienen una tendencia a estancarse en mínimos locales y -especialmente- en mesetas [20]. Dado lo anterior y considerando que ambos métodos sólo se centran en la optimización de los pesos (recordemos que a mayor profundidad de una red, mayor es su eficiencia [14]) es necesario explorar otras alternativas que permitan optimizar una DNN en todas sus dimensiones.

En la búsqueda de alternativas a BP, Engel [21] propuso en la década de 1980 optimizar los pesos de una FNN aplicando la técnica del recocido simulado (*simulated annealing*, SA). Casi simultáneamente, Montana y Davis [22] propusieron utilizar un algoritmo genético (GA) para optimizar los pesos de una FNN. Los buenos resultados obtenidos impulsaron la investigación en la materia (metaheurísticas aplicadas a la optimización de ANN) [2], [23], incluyendo tanto soluciones para la optimización de los pesos [24], [25], como para la optimización de la arquitectura y los pesos de manera simultánea [26], [27]. Asimismo se desarrolló un nuevo paradigma llamado *neuroevolución* (*neuroevolution*) [6], el cual considera la evolución de los componentes de la NN como parte fundamental de la misma, además del aprendizaje, dando origen a las *ANN evolutivas*.

Sin embargo, el surgimiento de las DNN, sumado al creciente volumen de datos, y a la heterogeneidad y sofisticación de los dispositivos tecnológicos que depara la *cuarta revolución industrial*, ha abierto la puerta a nuevas oportunidades para la optimización de las DNN. Es necesario contar con DNN simples, es decir es necesario optimizar los pesos y la arquitectura, y *adaptables*, ya que las necesidades e incluso las fuentes de datos cambian, así como con métodos de optimización que sean *eficientes*, en términos de su capacidad de actuar con la menor cantidad de datos y en el menor tiempo. Por todo esto, y en atención a las limitaciones de BP (principalmente a que sólo optimiza los pesos) y a los prometedores resultados obtenidos por las metaheurísticas en esta problema, con las condiciones actuales [25], estas últimas se presentan como buenas candidatas para aprovechar las oportunidades.

III. PROPUESTA: NEUROEVOLUCIÓN PROFUNDA

Considerando las oportunidades expuestas y el estado del arte, proponemos abordar el problema de la hiperparametriza-

ción de DNN mediante la DN, prestando especial atención en las RNN (por ser intrínsecamente las ANN más profundas).

Definimos como objetivo del trabajo de tesis el desarrollo de una técnica de DN *híbrida*, es decir que utilice tanto métodos *tradicionales* (como BP), como técnicas basadas en metaheurísticas. De este modo buscamos aprovechar las bondades de ambas estrategias, con el fin de obtener el mejor resultado posible.

Particularmente, centraremos nuestro trabajo en el estudio de las propiedades intrínsecas de las RNN, esto es buscaremos una manera simple de caracterizar la arquitectura de una red (configuración concreta de hiperparámetros que no considera los valores de los pesos), que permita comparar dos configuraciones de manera tal que sea posible determinar cuál de ellas se adapta mejor a un problema dado. Luego, usaremos esta información como entrada (la “heurística”) para definir un algoritmo metaheurístico de optimización de RNN. Finalmente, una vez escogida la mejor red para el problema, proponemos entrenar esta red con métodos del estado del arte. Es importante notar que de manera tentativa no limitamos el entrenamiento a BP ni al uso de metaheurísticas, sino que buscaremos la mejor forma de resolver el problema.

La siguiente lista enumera los objetivos principales del presente trabajo doctoral:

- G1 Definir un método para caracterizar RNN que permita comparar su desempeño esperado sin la necesidad de realizar un entrenamiento de la misma.
- G2 Diseñar e implementar una técnica de hiperparametrización de RNN basada en metaheurísticas que utilice la caracterización definida previamente.
- G3 Aplicar la técnica de hiperparametrización de RNN definida a la resolución de problemas en el contexto de SC.
- G4 Difundir los resultados mediante el desarrollo (y la puesta a disposición pública) de una librería software.

Cabe señalar que como parte de las directrices de este trabajo, proponemos utilizar la herramienta *TensorFlow* [28], una librería de código abierto para el aprendizaje de máquina utilizada (y desarrollada inicialmente) por Google. La idea de fondo es, además de reutilizar la implementación de la DNN, maximizar el impacto del objetivo G4.

IV. METODOLOGÍA Y PLAN DE TRABAJO

Para cumplir con los objetivos propuestos planteamos realizar un estudio teórico y empírico de la hiperparametrización de una RNN. Además de diseñar y construir artefactos de software que permitan implementar los algoritmos definidos, para luego aplicarlos a la resolución de problemas reales en los dominios establecidos.

Este trabajo tendrá una duración planificada de 36 meses y se definen 6 paquetes de trabajo (WP) para cumplir con los objetivos definidos. A continuación se describen los WP del proyecto:

- WP0 Gestión del Proyecto: Actividades relacionadas al control y planificación del proyecto, así como las

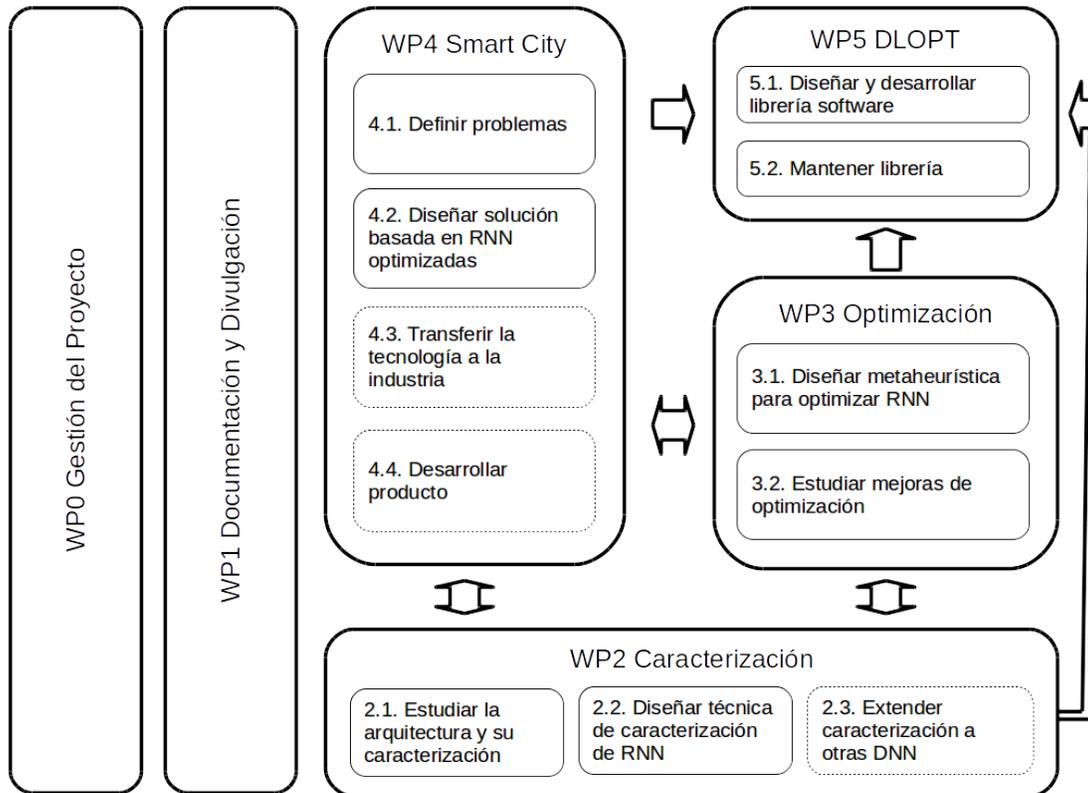


Figura 2. Paquetes de trabajo, sus actividades principales y relaciones.

actividades burocráticas requeridas para su presentación.

- WP1 Documentación y Divulgación: Generación de la tesis doctoral y actividades relativas a la promoción de los resultados.
- WP2 Caracterización: Actividades conducentes al desarrollo de una caracterización de la arquitectura de una RNN (objetivo **G1**).
- WP3 Optimización: Desarrollo de una técnica para la optimización de los hiperparámetros de una RNN basada en metaheurísticas y en la caracterización definida (objetivo **G2**).
- WP4 Smart City: Agrupación de actividades relativas al estudio de tres problemas reales en el contexto de la SC y su resolución mediante el uso de una RNN optimizada con la técnica definida (objetivo **G3**). Como parte de esta actividad se definirán los problemas concretos que serán abordados.
- WP5 DLOPT: Actividades relacionadas al desarrollo y mantenimiento de una librería software que incorpore los resultados del trabajo (objetivo **G4**).

Los paquetes **WP0** y **WP1** son funciones de apoyo a la tesis doctoral y por tanto su ciclo de vida está ligado a la duración del mismo. La Figura 2 muestra los paquetes de trabajo y sus actividades principales, así como la relación que existe entre los diferentes paquetes. Las actividades que aparecen con líneas punteadas son consideradas optativas para

la consecución del objetivo.

La Figura 3 muestra la planificación de alto nivel, incluyendo los hitos más relevantes de cada paquete de trabajo (destacados en color naranja en el calendario). La Sección IV-A muestra con un mayor nivel de detalle la planificación del paquete de trabajo **WP1**.

IV-A. Diseminación de los resultados

Con el fin de difundir los resultados de este trabajo y con el objetivo de establecer nexos que permitan contribuir a los mismos se plantea publicar artículos científicos en revistas especializadas, asistir y publicar en congresos relevantes, y desarrollar una página web temática.

Consideraremos al menos las siguientes revistas para la publicación de los resultados obtenidos (entre paréntesis se muestra la clasificación y calificación en el ranking JCR¹): Engineering Applications of Artificial Intelligence (Computer Science, Artificial Intelligence, Q1), Applied Soft Computing (Computer Science, Software, Q1) y Energies (Energy and Fuels., Q2).

Asimismo tendremos en cuenta los siguientes congresos: Congress on Evolutionary Computation (CEC, IEEE), The Genetic and Evolutionary Computation Conference (GECCO, ACM), EvoStar y Conference on Neural Information Processing Systems (NIPS, IEEE).

¹<http://www.scimagojr.com> [Accedido: 22-Agosto-2017]

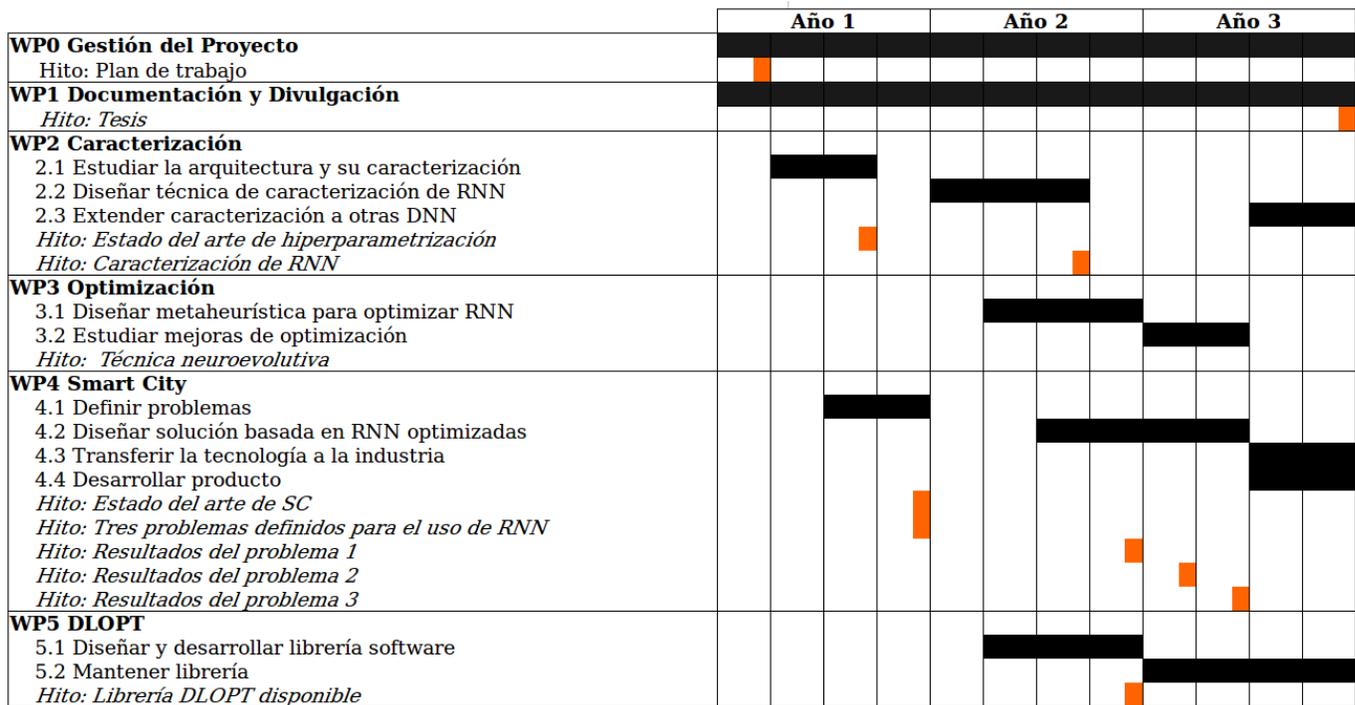


Figura 3. Plan de alto nivel. En negro se destacan los hitos más relevantes del trabajo.

Por otra parte, el sitio web temático será publicitado a través de los medios de difusión del grupo de investigación.

A la fecha, se cuenta con 3 artículos aceptados en conferencias [3], [29], [30] y 2 artículos en revisión (ya están disponibles de manera pública como *preprints*) [31], [32] relacionados con la propuesta de caracterización de RNN y su uso. Asimismo se han publicado 2 artículos en revistas de calidad [33], [34] y 1 en un congreso internacional [35] relacionados a los problemas de SC escogidos (pero no ligados a RNN). A estos trabajos se suman otros trabajos en progreso y en revisión (no disponibles públicamente).

V. RELEVANCIA

Este trabajo se centra en la resolución de un problema concreto: la hiperparametrización de una RNN, que si bien ya cuenta con múltiples soluciones propuestas, todavía es posible avanzar en la obtención de mejores resultados [2]. Asimismo, proponemos utilizar nuestros avances para la resolución de tres problemas en el contexto de las SC. Por este motivo, y dada la naturaleza y la relevancia de los problemas concretos [36], se espera que los resultados de este trabajo tengan un impacto positivo en la sociedad.

Esperamos también que los resultados de este trabajo, especialmente lo referido a los objetivos G1 y G2, sean de utilidad para la resolución del problema en otros tipos de DNN. Lo cual permitiría aumentar el impacto del trabajo considerablemente. Por este motivo, dependiendo del tiempo disponible y de los resultados obtenidos, planteamos nuestra intención de investigar en esa línea.

Por otra parte, el concepto de DN es reciente, por lo cual esperamos contribuir a su cimentación. No solo haciendo uso del conocimiento actual, sino que también aprovechando la experiencia del equipo en neuroevolución [23].

Finalmente, en lo referido al impacto más tangible de este trabajo (objetivo G4), esperamos contribuir activamente a la mejora de los resultados de los diversos usos que hoy en día se hacen de las RNN. Lo cual será una especie de efecto colateral de nuestro trabajo.

VI. CONCLUSIONES

En este trabajo hemos presentado nuestra propuesta de tesis doctoral, la cual busca abordar el problema de la hiperparametrización de redes neuronales recurrentes mediante la neuroevolución profunda.

Más allá de los resultados concretos de este trabajo, una tesis doctoral y los artefactos relacionados, creemos que la consecución de los mismos permitirá formar un investigador de alta calidad. Este macro resultado puede ser revisado desde tres ámbitos diferentes: personal, científico e industrial.

En el ámbito **personal** se espera desarrollar las habilidades necesarias para comprender de manera sistemática un campo de estudio, así como también desarrollar la capacidad de realizar una investigación de calidad en dicho campo. De forma puntual se espera validar estas habilidades mediante el aprendizaje de los campos relativos a la tesis doctoral. Asimismo se espera desarrollar la habilidad de analizar críticamente, evaluar y sintetizar nuevas y complejas ideas. También se espera impulsar la capacidad de integración en equipos

multidisciplinares y multiculturales, mediante el trabajo cotidiano en el entorno del grupo de investigación y fomentando la estancia en centros externos.

Por otra parte, se espera que al finalizar el periodo de formación se habrá desarrollado la habilidad de comunicación con la comunidad académica y científica, y con la sociedad en general. Lo cual será indispensable para una correcta difusión de los resultados obtenidos en este trabajo.

En términos generales se espera formar un investigador que sea capaz de aportar a la sociedad mediante el uso de sus habilidades y conocimientos para la resolución de problemas, lo cual sin lugar a dudas tiene un impacto positivo.

En el ámbito **científico** se espera contribuir al estado del arte, lo cual se validará a través de la consecución de publicaciones de calidad.

Finalmente, en el ámbito **industrial** se espera contribuir a solucionar problemas en el contexto de la ciudad inteligente idealmente se espera canalizar estos aportes a través de la transferencia tecnológica a empresas que enfrenten dichos problemas o mediante la creación de un producto comercial. Si bien este punto es de suma complejidad, lograr aportar en él impactaría de forma positiva a la economía, y lo que es más importante, podría significar un aporte a nuestras ciudades.

REFERENCIAS

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 60, no. January, pp. 97–116, 2017.
- [3] A. Camero, J. Toutouh, and E. Alba, "Comparing deep recurrent networks based on the mae random sampling, a first approach," in *Conference of the Spanish Association for Artificial Intelligence (CAEPIA) 2018*. Springer, 2018, pp. 1–10.
- [4] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat, "Evolving Deep Neural Networks," *arXiv preprint arXiv:1703.00548*, 2017. [Online]. Available: <http://arxiv.org/abs/1703.00548>
- [5] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, "Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning," *arXiv preprint arXiv:1712.06567*, 2017. [Online]. Available: <http://arxiv.org/abs/1712.06567>
- [6] X. Yao, "A review of evolutionary artificial neural networks," *International journal of intelligent systems*, vol. 8, no. 4, pp. 539–567, 1993.
- [7] S. Haykin, *Neural Networks and Learning Machines*, 2008, vol. 3.
- [8] M. A. Arbib, *The metaphorical brain 2: Neural networks and beyond*. John Wiley & Sons, Inc., 1989.
- [9] M. W. Dubin, *How the brain works*. John Wiley & Sons, 2013.
- [10] W. S. McCulloch and W. Pitts, "A Logical Calculus of the Idea Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.
- [11] S.-i. Amari, "Backpropagation and Stochastic Gradient Descent Method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.
- [12] Y. LeCun, "Une procédure d'apprentissage pour réseau a seuil asymétrique (a learning scheme for asymmetric threshold networks)," in *Proceedings of Cognitiva 85, Paris, France*, 1985.
- [13] D. Rumelhart, G. E. Hinton, and R. j. Williams, "Learning Internal Representations by Error Propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep. No. ICS-8506, 1985.
- [14] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks," *Advances in Neural Information Processing Systems*, vol. 19, no. 1, p. 153, 2007.
- [15] G. E. Hinton, S. Osindero, and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [16] G. Bebis and M. Georgiopoulos, "Feed-Forward Neural Networks: Why Network Size is so Important," *IEEE Potentials*, pp. 27–31, 1994.
- [17] A. M. Ukrainec and S. Haykin, "A Modular Neural Network for Enhancement of Cross-Polar Radar Targets," *Neural Networks*, vol. 9, no. 1, pp. 143–167, 1996.
- [18] F. Gruau, L. C. B.-I. I, O. A. D. D. Doctorat, M. J. Demongeot, E. M. M. Cosnard, M. J. Mazoyer, M. P. Peretto, and M. D. Whitley, "Neural Network Synthesis Using Cellular Encoding And The Genetic Algorithm," Ph.D. dissertation, 1994.
- [19] M. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, p. 409, 1952.
- [20] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization," in *Advances in neural information processing systems*, 2014, pp. 2933–2941.
- [21] J. Engel, "Teaching Feed-Forward Neural Networks by Simulated Annealing," *Complex Systems*, vol. 2, pp. 641–648, 1988.
- [22] D. J. Montana and L. Davis, "Training Feedforward Neural Networks Using Genetic Algorithms," *Proceedings of the 11th International Joint Conference on Artificial intelligence - Volume 1*, vol. 89, pp. 762–767, 1989.
- [23] E. Alba and R. Martí, *Metaheuristic Procedures for Training Neural Networks*. Springer Science & Business Media, 2006, vol. 35.
- [24] D. Whitley, T. Starkweather, and C. Bogart, "Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity," *Parallel Computing*, vol. 14, no. 3, pp. 347–361, 1990.
- [25] G. Morse and K. O. Stanley, "Simple evolutionary optimization can rival stochastic gradient descent in neural networks," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*. ACM, 2016, pp. 477–484.
- [26] T. B. Luderemir, A. Yamazaki, and C. Zanchettin, "An optimization methodology for neural network weights and architectures," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1452–1459, 2006.
- [27] M. T. Mahmoudi, F. Taghiyareh, N. Forouzideh, and C. Lucas, "Evolving Artificial Neural Network Structure Using Grammar Encoding and Colonial Competitive Algorithm," *Neural Computing and Applications*, vol. 22, no. 1, pp. 1–16, 2013.
- [28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *OSDI*, vol. 16, 2016, pp. 265–283.
- [29] A. Camero, J. Toutouh, D. H. Stolfi, and E. Alba, "Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities," in *Learning and Intelligent Optimization Conference LION*, 2018.
- [30] A. Camero, J. Toutouh, J. Ferrer, and E. Alba, "Waste generation prediction in smart cities through deep neuroevolution," in *Congreso Iberoamericano de Ciudades Inteligentes (ICSC-CITIES 2018)*, 2018.
- [31] A. Camero, J. Toutouh, and E. Alba, "Low-cost recurrent neural network expected performance evaluation," *arXiv preprint arXiv:1805.07159*, may 2018.
- [32] A. Camero, J. Toutouh, and E. alba, "DLOPT: Deep learning optimization library," *arXiv preprint arXiv:1807.03523*, july 2018.
- [33] A. Camero, G. Luque, Y. Bravo, and E. Alba, "Customer Segmentation Based on the Electricity Demand Signature : The Andalusian Case," *Energies*, vol. 11, no. 7, p. 1788, 2018.
- [34] A. Camero, J. Arellano-Verdejo, and E. Alba, "Road map partitioning for routing by using a micro steady state evolutionary algorithm," *Engineering Applications of Artificial Intelligence*, vol. 71, pp. 155–165, 2018.
- [35] A. Camero, J. Arellano-Verdejo, C. Cintrano, and E. Alba, "Tile map size optimization for real world routing by using differential evolution," in *2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings*, 2017.
- [36] "EU Commision: Horizon 2020 - The EU Framework Programme for Research and Innovation." <https://ec.europa.eu/programmes/horizon2020>, accessed: 2017-07-12.