1. Datos personales del alumno.

Nombre: Daniel

Apellidos: García-Ocaña Hernández

DNI: 47294991 E

Domicilio: C/Alfredo Marqueríe, portal 5 izquierda, piso 2ºB

Número de teléfono: 660 19 22 70

E-mail: d.garcia-ocana@alumnos.upm.es

Estudios que realiza: Máster Universitario en Ingeniería de Telecomunicación.

2. Entidad colaboradora donde se han realizado las prácticas y lugar de ubicación.

Entidad colaboradora/unidad de acogida: Señales, Sistemas y Radiocomunicaciones.

Ubicación: Escuela Técnica Superior de Ingenieros de Telecomunicación (Av. Complutense, 30, 28040 Madrid), edificio C, laboratorio 303.

3. Descripción concreta y detallada de las tareas, trabajos desarrollados y departamentos de la entidad a los que ha estado asignado.

Descripción concreta y detallada de las tareas y trabajos desarrollados: A lo largo de este período (01.02.2017 – 30.04.2017) se ha llevado a cabo una completa formación en los campos de aprendizaje por refuerzo o reinforcement learning (RL) y aprendizaje profundo o deep learning (DL). Con el objetivo de adquirir una base teórica sólida, se realizó una revisión de los fundamentos de optimización convexa, teoría de la dualidad y teoría de control óptimo, seguida de la formación en todo lo concerniente a redes neuronales artificiales y su aplicación al problema de aprendizaje por refuerzo. Una vez se dispuso de la base teórica apropiada, se entró en una etapa de investigación pura para aplicar todos estos conocimientos al desarrollo de nuevos algoritmos de aprendizaje por refuerzo, objetivo final de esta beca. De este modo, los hitos que se marcaron fueron los siguientes:

A.1) Introducción al campo del aprendizaje por refuerzo: a través de [1] se llevó a cabo la primera toma de contacto con los conceptos relativos al aprendizaje por refuerzo (formulación del problema de aprendizaje por refuerzo, programación dinámica (policy y value iteration), métodos de Monte Carlo, métodos de predicción “Temporal-Difference” y métodos de control “SARSA” y “Q-learning”, etc…). Con el objetivo de asentar mejor las bases de estos métodos, se siguió además el video-curso online “Reinforcement Learning” proporcionado por el “Georgia Institute of Technology”. Al mismo tiempo que se avanzaba con los conocimientos teóricos, se realizaron una serie de ejercicios que aparecen propuestos en [1] para comprobar el correcto entendimiento de todos los conceptos. (**Fin hito A.1: 01.02.2017 – 15.02.2017**)

A.2) Introducción a los algoritmos existentes de aprendizaje por refuerzo: con [2] se tomó una perspectiva más completa del aprendizaje por refuerzo descubriendo la importancia que toman el operador de Bellman y las cadenas de Markov. De este modo, se trata de una interpretación más técnica de la que se encontró en [1], con la que además tuve un primer acercamiento al tipo de algoritmos con los que se trabajará. Para completar este enfoque, se empleó además [3], donde se sintetizan los conceptos de la programación dinámica y el aprendizaje por refuerzo y se muestran además implementaciones de los algoritmos más empleados a día de hoy: SARSA y Q-learning. Con el objetivo de comprobar el correcto entendimiento de toda la literatura leída tanto en el hito A.1 como en el A.2, se llevó a cabo la implementación de los dos algoritmos

anteriormente citados, de manera que resolviesen 3 problemas tipo que aparecen en [1]: gridworld, cliff problem y random walk. (**Fin hito A.2: 15.02.2017 – 18.02.2017**)

Con la realización de ejercicios citada en el hito A.1 así como con la implementación de los algoritmos SARSA y Q-learning para algunos problemas tipo, se dio por finalizada la primera etapa de aprendizaje de los fundamentos básicos sobre aprendizaje por refuerzo.

B.1) Revisión de los fundamentos de optimización convexa: a través de [4], se formó una base teórica sólida en lo que a los conceptos de convexidad y optimización se refiere (conjuntos y funciones convexas, problemas de optimización convexos y no convexos, teoría dual y problema dual de Lagrange, etc…). Para afianzar los conocimientos obtenidos, se siguió además el video-curso online "Convex Optimization" proporcionado por la universidad de Stanford impartido por el profesor Stephen Boyd, autor de [4]. Como se verá más adelante, en este primer hito cobrará especial relevancia la teoría dual. De nuevo, al mismo tiempo que se avanzaba con los conocimientos teóricos, se realizaron una serie de ejercicios que aparecen propuestos en [4] para comprobar el correcto entendimiento de todos los conceptos. (**Fin hito B.1: 18.02.2017 – 04.03.2017**)

Teniendo ya una base teórica apropiada en aprendizaje por refuerzo y optimización convexa, se pasó a una literatura que combina ambos conceptos orientándolos a la teoría del control óptimo:

C.1) Estudio de la conexión entre el problema de control óptimo y su formulación como un problema de programación lineal: cuando se conocen los datos del proceso de decisión de Markov que modela nuestro problema, se puede hacer uso de programación dinámica para encontrar la política de comportamiento óptima. En [5] se presenta un enfoque alternativo al de la programación dinámica para la resolución del problema de aprendizaje por refuerzo basada en programación lineal, donde la variable a optimizar es la función valor. Esta nueva visión, junto con la teoría dual mencionada en el hito B.1, será la base de desarrollo de los nuevos algoritmos buscados. (**Fin hito C.1: 05.03.2017 – 09.03.2017**)

Llegados a este punto, es importante mencionar que existen dos clases de métodos que permiten resolver el problema de toma de decisiones/problema de control óptimo/problema de aprendizaje por refuerzo: los basados en un modelo del entorno y aquellos en los que se carece de un modelo del mismo. En principio, la visión que nos va a interesar es aquella en la que no se conoce un modelo del sistema, pues será el caso más común. En la actualidad, existen una serie de algoritmos de aproximación estocástica que permiten resolver ecuaciones de punto fijo (como las ecuaciones de Bellman, que resuelven el problema de control óptimo) en base a la experiencia generada conforme avanza el tiempo. Entre ellos cabe destacar SARSA o Q-learning, algoritmos muy aceptados en la comunidad del aprendizaje por refuerzo que permiten una implementación estocástica eficaz y eficiente.

El trabajo de investigación trata, por tanto, de la búsqueda de nuevos algoritmos que sean capaces de explotar la experiencia generada a lo largo del tiempo, haciendo uso de la formulación como un programa lineal que se expone en [5]. Y es en este punto donde toma especial relevancia la teoría dual mencionada en el hito B.1.

D.1) Búsqueda de algoritmos con un enfoque primal-dual que permitan la reformulación del programa lineal expuesto en [5] para la resolución del problema de optimización de la función de coste (función valor): En C.1 se presentó la idea de resolver el problema de control óptimo a través de un programa lineal. A partir de la teoría dual, sabemos que dicho programa lineal se puede resolver también a través de su problema dual, y más aún, que avanzar hacia la solución óptima en el problema primal implica avanzar hacia la solución óptima en el problema dual. Con estas

premisas, a través de [6] se estudió la posibilidad de formular el problema de control óptimo como un problema de saddle point (o de punto de silla) del Lagrangiano, y es en este punto donde comienzan las aportaciones de toda la formación expuesta en los hitos anteriores. Para solucionar este problema de saddle point, se evaluaron dos técnicas de gradiente[1] ampliamente conocidas y utilizadas: (1) el método propuesto por Arrow-Hurwicz, que tras varias pruebas y consultas a diversas referencias se pudo comprobar que no era válido para nuestro caso, y (2) el método dual-ascent, con el cual sí se encontraron pruebas de convergencia.

Con este nuevo enfoque basado en la teoría dual, se demuestra en [6] que de la variable dual se puede extraer la política de comportamiento del problema de control bajo estudio, con lo cual, resolver el problema de saddle point encontrando la variable dual óptima supone encontrar la política óptima. Es decir, el nuevo método dual planteado permite buscar la solución óptima en el espacio de políticas en lugar de en el espacio de las funciones valor tal y como hacen los conocidos algoritmos SARSA y Q-learning.

En vista de la aparente idoneidad de este nuevo planteamiento desarrollado, se realizaron dos implementaciones del método dual-ascent para el problema de saddle point que resuelve el problema de control planteado:

1) una que resolviera el problema supuesto conocido el modelo del sistema, con el objetivo de verificar la validez del algoritmo y su convergencia a la solución óptima:
   - para la minimización de la variable primal se propuso resolver las ecuaciones de Bellman de manera analítica, de modo que se obtuviera la función valor óptima en cada iteración para la política actual (definida por la variable dual).
   - para la maximización de la variable dual se empleó ascenso por gradiente, generando en cada iteración una nueva política.
2) otra que permitiera tener en cuenta la experiencia generada, obtenida del entorno:
   - para la minimización de la variable primal se empleó TD, método de aproximación estocástica que en cada actualización itera sobre la ecuación de Bellman, generando un punto factible del problema de saddle point formulado. Esta etapa correspondería a la evaluación de la política.
   - para la maximización de la variable dual se empleó ascenso por gradiente estocástico, de manera que se pudiera actualizar la política (o lo que es lo mismo, la variable dual) con cada muestra de experiencia obtenida.

En ambos casos, cuando el algoritmo ha convergido la variable dual ya no se actualiza (representa la política estacionaria óptima del problema planteado), y en consecuencia la variable primal tampoco (convergencia de la ecuación de punto fijo, función valor óptima).

Tras comprobar el correcto funcionamiento de ambas implementaciones con un MDP relativamente sencillo, se pasó a enfrentar nuestro algoritmo con SARSA y Q-learning para dos problemas típicos: random walk y el problema del acantilado (cliff walking). Con intención de encontrar fortalezas y debilidades tanto de nuestro algoritmo como de Q-learning y SARSA, se probaron los dos problemas mencionados anteriormente en los casos en que (1) la matriz de transición es determinista y (2) la matriz de transición es aleatoria (probabilidad 0.8 de tomar la acción deseada, probabilidad 0.2 de tomar otra acción diferente). La conclusión que se extrajo de estas pruebas fue que nuestro algoritmo es más sensible a la exploración que sus otros dos competidores. En consecuencia, consigue converger más rápido que SARSA y Q-learning en los

[1] El hecho de emplear métodos de gradiente se debe a que nos van a permitir desarrollar algoritmos de aproximación estocástica, mediante la aproximación del gradiente a través de las muestras de experiencia recogidas.

casos en que la matriz de transición es aleatoria. Por contraparte, cuando la matriz de transición es determinista la exploración es menor y los métodos tradicionales, SARSA y Q-learning, proporcionan mejores resultados de convergencia.

Para evaluar la convergencia de los algoritmos, lo que se hizo fue comparar la recompensa instantánea acumulada promediada en varios experimentos, empezando siempre en el mismo estado. De este modo, el algoritmo que convergía en menos episodios a la máxima recompensa posible era el de mejores propiedades de convergencia.

Así termina el hito D.1, con la formulación, prueba y evaluación de un nuevo algoritmo de aprendizaje por refuerzo basado en la teoría dual. (**Fin hito D.1: 09.03.2017 – 26.03.2017**)

Todas las tareas realizadas en el hito D.1 aplican únicamente al caso en que el problema es de pequeña escala; es decir, cuando el número de estados es discreto y finito. Del día 29.03.2017 hasta el final de la beca, lo que se hizo fue adaptar la formulación y el algoritmo para funcionar en problemas de mayor complejidad en los que el espacio de estados es muy grande o continuo (y por tanto ya no podemos almacenar una función valor por cada estado). Más concretamente se adaptó el algoritmo para ser capaz de resolver problemas en los que se tiene:

a) una aproximación lineal de la función valor/de coste, representada por el producto de un vector de características por otro de pesos asociados a cada característica.
b) una aproximación no lineal de la función valor/de coste, representada por una red neuronal en la que el vector de características será ahora la red neuronal en sí y los pesos asociados a cada interconexión de las neuronas serán el "tensor" de características.

E.1) Resolución de problemas por refuerzo con el nuevo algoritmo desarrollado mediante la aproximación lineal de la función valor: a través de [1] y [7] se estudió la problemática que surge cuando el espacio de acciones es muy grande o continuo, y la manera de resolverlo aproximando de manera lineal la función valor que permite obtener la política de comportamiento óptima de nuestro problema. Se hizo una revisión de las familias de funciones base comúnmente empleadas para la definición del vector de parámetros, y posteriormente, de los algoritmos desarrollados a partir de esta formulación mediante aproximaciones lineales: LSPI, LSTD y GTD2. Teniendo clara la teoría y aplicación de dichas aproximaciones de la función valor, se adaptó esta interpretación a nuestro problema de saddle point que resuelve el problema de control, primero introduciendo este nuevo concepto al programa lineal original, tal y como se detalla en [8], y a continuación derivando el Lagrangiano asociado. Como resultado, se obtuvo un nuevo algoritmo basado en dual-ascent y aproximaciones lineales capaz de funcionar cuando el espacio de estados es muy grande o continuo.

Una vez más, se enfrentó nuestro nuevo algoritmo con otro considerado ya una tecnología "madura", LSPI (Least-Squares Policy Iteration), para dos problemas típicos: chain walk y el problema del coche en la montaña (mountain car). El primero de ellos, con únicamente 4 estados, sirvió de prueba de concepto para verificar los desarrollos anteriores y la correcta implementación. El segundo, mountain car, supuso una prueba más realista ya que se trata de un problema en el que el espacio de estados es continuo. Una vez más, nuestro algoritmo fue capaz de resolver ambos problemas, demostrándose de nuevo que funciona mejor cuando las transiciones contienen un ligero carácter aleatorio, y por tanto exploratorio.

En la actualidad, principalmente se usan aproximaciones lineales para aproximar la función valor, pues no existe una demostración matemática de que se pueda aproximar mediante una función cualquiera no lineal. No obstante, sí hay pruebas empíricas y por tanto a veces se usan aproximaciones no lineales. Como línea de investigación transversal a este hito se planteó la

demostración desde el enfoque primal-dual de que cualquier función, lineal o no, puede aproximar la función valor. Aunque casi conseguida esta demostración, a día de hoy falta por formalizar algunas premisas consideras.

Una tarea futura que se planteó a raíz de este hito fue la prueba de aproximar la variable dual también de manera lineal con el objetivo de mejorar los resultados de convergencia obtenidos. (**Fin hito E.1: 27.03.2017 – 07.04.2017**)

E.2) Resolución de problemas por refuerzo con el nuevo algoritmo desarrollado mediante la aproximación no lineal de la función valor, en concreto, mediante redes neuronales artificiales: de nuevo, la problemática a resolver es la misma que en el hito E.1 pero en este caso se aborda la resolución del problema haciendo uso de redes neuronales artificiales. Este punto de vista, relativamente nuevo, que combina aprendizaje por refuerzo y redes neuronales es lo que actualmente se conoce como aprendizaje por refuerzo profundo o deep reinforcement learning (DRL). Este hito por tanto se pudo dividir en cuatro etapas:

1. Formación en redes neuronales y estructuras profundas: a través de [9] y diversas fuentes de Internet, se llevó a cabo una rápida formación en redes neuronales y los diferentes tipos de arquitecturas (profundas y no profundas) y tipos de redes (FNN, RNN, CNN, etc..) existentes y empleadas en la actualidad
2. Formación en la implementación de redes neuronales: tras la obtención de la base teórica, se aprendió el manejo con los frameworks y librearías más comúnmente empleadas en la industria para la implementación de redes neuronales, así como su integración en Python:
   - Tensorflow: por su gran comunidad (y soporte) en general y por su uso por parte de las soluciones desarrolladas por Google en particular.
   - Keras: nos permite una capa de abstracción sobre TensorFlow que, como se pudo ver, facilita la implementación de las redes neuronales.

   Para tener una primera toma de contacto con estas herramientas, se siguió el tutorial clásico de iniciación en la implementación de redes neuronales: clasificación de números (obtenidos del conjunto de datos de entrenamiento MNIST) escritos a mano. Una vez familiarizado con estas librerías, se implementó un sencillo modelo de regresión lineal para comprobar los conocimientos aprendidos.
3. Unión de los conceptos de aprendizaje por refuerzo y aprendizaje profundo: tras tener una base sólida en RL y DL se estudió en [7] la unión de ambos conceptos en lo que se conoce como aprendizaje por refuerzo profundo; ahora, la función valor del problema de control pasará a estar representada por una red neuronal. Además, se llevó a cabo una revisión del estado del arte de este campo, conociendo así las soluciones más empleadas a día de hoy: Neural Fitted Q-iteration (NFQ) y Deep Q-learning networks (DQN).
4. Desarrollo del nuevo algoritmo basado en aproximaciones no lineales mediante redes neuronales: teniendo clara la teoría y aplicación de dichas aproximaciones de la función valor mediante redes neuronales, se adaptó esta interpretación a nuestro problema de saddle point que resuelve el problema de control, primero introduciendo este nuevo concepto al programa lineal original, y a continuación derivando el Lagrangiano asociado.

Como resultado de estas 4 etapas, se obtuvo un nuevo algoritmo basado en dual-ascent y aproximaciones no lineales mediante redes neuronales capaz de funcionar cuando el espacio de estados es muy grande o continuo.

Si bien es cierto que se realizó la implementación de este algoritmo y se probó de manera exitosa con los dos problemas típicos anteriores, chain walk y el problema del coche en la montaña, no

se dispuso de tiempo para enfrentar dichos resultados con los de Neural Fitted Q-iteration o Deep Q-learning networks. Por tanto, se deja esta labor como tarea futura posterior a la beca. (**Fin hito E.2: 08.04.2017 – 30.04.2017**)

En este punto finalizan las tareas desarrolladas a lo largo de la beca.

Departamento de la entidad en el que ha estado asignado: Grupo de Aplicaciones de Procesado de Señales.

Referencias

[1] R. S. Sutton y A. G. Barto, Reinforcement Learning: an introduction, The MIT Press, 2015.

[2] C. Szepesvari, Algorithms for Reinforcement Learning, Morgan & Claypool Publishers, 2009.

[3] L. Busoniu, R. Babuska, B. De Schutter y D. Ernst, Reinforcement learning and dynamic programming using function approximators.

[4] S. Boyd y L. Vandenberghe, Convex Optimization, Cambridge University Press.

[5] D. P. Bertsekas, Dynamic Programming and Optimal Control, Athena Scientific, 2012.

[6] T. Wang, D. Lizotte, M. Bowling y D. Schuurmans, Dual Representations for Dynamic Programming, Journal of Machine Learning, 2008.

[7] S. Zazo Bello y S. Valcarcel Macua, Notes on Reinforcement Learning, Master in Signal Theory and Communications, 2017.

[8] B. Van Roy y D. P. De Faricas, «The linear programming approach to approximate dynamic programming,» 2002.

[9] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning, MIT Press, 2016.

4. Valoración de las tareas desarrolladas con los conocimientos y competencias adquiridos en relación a los estudios universitarios.

A lo largo del máster, únicamente se trataron conceptos relativos a la temática de estas prácticas en una asignatura del primer cuatrimestre del primer curso, análisis de señal para comunicaciones. De hecho, fue la asignatura que reavivó en mí el interés sobre lo relativo a la teoría de optimización.

Dado que estas prácticas me han permitido poner a prueba las nociones que se aprendieron en dicha asignatura, y ampliar enormemente mis conocimientos en todo el campo de la optimización y la inteligencia artificial en general, considero que las tareas desarrolladas a lo largo de estas prácticas han sido idóneas. En resumen:

1. Me han permitido repasar y llevar a la práctica conceptos teóricos adquiridos en el máster, así como enfrentarme a problemáticas distintas/nuevas
2. Me han permitido crecer en el ámbito profesional, aprendiendo mucho sobre la rama del aprendizaje por refuerzo, que hace un uso importante de la teoría de optimización ya estudiada.

Si bien es cierto que en ocasiones algunas tareas resultaban inicialmente muy complicadas por falta de base en algún campo matemático, al final siempre he conseguido superar “el obstáculo” y sacar adelante la tarea, bien sólo o con ayuda del resto de miembros del departamento.

5. Relación de los problemas planteados y el procedimiento seguido para su resolución.

El principal problema planteado ha sido el desarrollo de un nuevo algoritmo orientado al aprendizaje por refuerzo que sea competitivo frente a los ya existentes y más conocidos, tales como SARSA, Q-learning, LSPI, DQN o NFQ entre los más destacables. Para ello, se pasó por el proceso natural de: (1) resolver el problema de control óptimo en problemas de pequeña escala cuando se conoce el modelo y cuando no se conoce el modelo, y (2) resolver el problema de control óptimo en problemas en los que no se conoce el modelo y el espacio de estados es muy grande o continuo

Para resolver este problema (es decir, encontrar el nuevo algoritmo), se llevó a cabo un estudio del estado del arte en lo que a reinforcement learning se refiere (hitos citados anteriormente), de manera que se concluyó que un método alternativo podría ser buscar en el espacio de políticas en lugar de en el de las funciones valor. Tras consultar varias referencias, se adoptó la interpretación del problema de control óptimo como un programa lineal. Con esta formulación, se pudo hacer uso de la teoría dual para resolver dicho problema de optimización (en el que la función objetivo es la función valor) a través de su formulación como un problema max-min de saddle point del Lagrangiano, en el cual la variable dual es la política a seguir (y la primal sigue siendo la función valor).

De este modo, se ha consiguió reformular el problema inicial expresado en forma de problema de optimización de una función valor, como un problema de optimización de la política a seguir (búsqueda en el espacio de políticas).

Para abordar la resolución del problema de saddle point, se recurrió primero al método propuesto por Arrow y Hurwicz. Al llevarlo a la práctica nos encontramos con que dicho algoritmo nunca convergía a la solución del saddle point, sino que era altamente oscilante. Consultando en la publicación en la que Arrow y Hurwicz presentan este método, se encontró que cuando la función objetivo es lineal, el algoritmo es altamente oscilante y no converge a menos que el punto de partida elegido se encuentre en una bola cercana al saddle point. Debido a este inconveniente, este primer método fue dejado de lado. Como solución alternativa, se pasó a probar el método dual-ascent, el cual se pudo comprobar que sí convergía al saddle point.

Llegados a este punto, se consiguió definir la “forma” que tendría nuestro nuevo algoritmo. El siguiente problema que surgió fue adaptarlo a problemas de pequeña y gran escala. Para problemas de pequeña escala, la adaptación fue instantánea. Para adaptarlo a problemas en los que el espacio de estados es muy grande o continuo hubo que documentarse sobre todo lo relativo a la aproximación de funciones, tanto lineales como no lineales.

La aproximación mediante funciones no lineales se realizó mediante redes neuronales, las cuales eran totalmente desconocidas para mí. Por ello, se trató de otro problema a abordar. Para resolverlo, se adquirió de nuevo una base teórica sólida en la materia y se siguieron diversos video-tutoriales en los que se explicaba cómo implementar redes neuronales con TensorFlow y Keras.

En lo que refiere a los problemas que surgieron a lo largo de los hitos desarrollados en el punto 3 de este documento, estos fueron de diverso carácter:

Problemas en la resolución de ejercicios de base: tras la obtención de la base teórica en cada hito, se llevaron a cabo una serie de ejercicios para comprobar el correcto entendimiento de los conceptos de mayor relevancia. En ocasiones alguna idea no me quedó completamente clara y en consecuencia no era capaz de resolver el problema en cuestión. De este modo, consultando a

los compañeros del grupo fui capaz de aclarar todas las ideas y resolver los problemas de carácter teórico.

Problemas de implementación: en la implementación de algoritmos, no siempre se obtenía el resultado esperado. Para solucionar este problema, se realizaba un proceso de depuración (debugging) en el cual se podían detectar tanto errores conceptuales (aspectos teóricos que pensé que me habían quedado claros, pero no era así) como de implementación. Me fue de gran utilidad la consulta de distintos foros que trataban estas temáticas en internet, así como la lectura de diversos artículos y más documentación que trataban aspectos de implementación.

6. Identificación de las aportaciones que, en materia de aprendizaje, han supuesto las prácticas, así como una valoración personal de la experiencia, en términos de aprendizaje.

Podría decir que estas prácticas me han tenido en un estado de aprendizaje continuo, pues a medida que avanzaba con la investigación, surgían problemas nuevos y generalmente desconocidos para mí. He ganado mucha base matemática en general, y en materia de optimización, teoría dual, teoría de control y aprendizaje por refuerzo en particular. He conocido de primera mano cómo se estructura la rama de aprendizaje por refuerzo y he podido tener un acercamiento muy práctico e interesante a las redes neuronales y su uso hoy en día. Al margen de los conocimientos puramente teóricos, he aprendido muchos aspectos importantes en lo que a la algorítmica necesaria para resolver los problemas planteados se refiere (el enfoque de programación dinámica y los conceptos relativos a los algoritmos de aproximación estocástica), así como como las diferencias conceptuales entre los algoritmos de aprendizaje por refuerzo que constituyen el estado del arte y todo lo concerniente a las métricas empleadas para la comparación de algoritmos de RL. En general, considero que las aportaciones de estas prácticas me han permitido tomar los conocimientos esenciales, y muchos avanzados, en lo que a data science e inteligencia artificial se refiere, de manera que ahora pueda seguir mi carrera profesional en esta disciplina.

Como competencia transversal, también he aprendido a desenvolverme en la presentación de mis resultados en las reuniones semanales. Considero también que he ganado vocabulario, expresividad y confianza en esta materia que era tan nueva para mí al comienzo de las prácticas.

Por último y no menos importante, he aprendido lo que conlleva trabajar en el ámbito investigador. Acostumbrado durante los años de grado y máster a poner en práctica cosas que ya sabía que funcionaban de antemano, en esta beca he podido ver de cerca lo que supone abordar un problema nuevo totalmente desconocido y la tarea que ello implica: definir alternativas de testeo para asegurarse de que se avanza de manera correcta, apoyarse en demostraciones matemáticas que avalen el procedimiento a implementar. En definitiva, seguir una metodología de trabajo. En lo que respecta a lo personal, también supone enfrentarse a situaciones en las que todo el tiempo y trabajo invertido en algo que creía que iba a funcionar no sale como esperaba, de manera que he aumentado la tolerancia a la frustración que suponen estas situaciones.

En resumen, creo que ha sido una experiencia muy enriquecedora en todos los aspectos (laborales, académicos y personales), ya que ha aumentado mi conocimiento sobre data science en general y reinforcement learning en particular, me ha permitido mejorar mis habilidades referidas a la programación, tanto en Matlab como en Python, orientadas a reinforcement learning y me ha aportado un enfoque del trabajo investigador que desconocía hasta el momento así como la importancia de mantenerse actualizado al estado del arte.

7. Evaluación de las prácticas y sugerencias de mejora.

Además de lo citado en el punto 6 de este documento, añadir que estoy completamente satisfecho con el desarrollo de estas prácticas ya que me han proporcionado un punto de vista más realista de lo que hay después del grado/máster y me han ayudado a decidir qué quiero.

Como sugerencia de mejora, únicamente mencionar que en ocasiones he notado que necesitaba más capacidad computacional para simular determinados problemas, por lo que creo que sería de gran utilidad, para todo aquel que trabaje actualmente o en el futuro en el departamento, disponer de al menos una GPU a la que se puedan mandar tareas en remoto.

8. Diario o registro de la actividad realizada.

Registro de actividad realizada, ordenada por orden cronológico:

1. Aprendizaje por refuerzo: el problema de control óptimo
   1.1. Introducción al campo del aprendizaje por refuerzo
   1.2. Introducción a los algoritmos existentes de aprendizaje por refuerzo cuando el número de estados es discreto y finito (problemas de pequeña escala)
   1.3. Puesta a prueba de los conocimientos adquiridos: problemas teóricos e implementación y prueba de los algoritmos estudiados
2. Optimización convexa
   2.1. Revisión de los fundamentos de optimización convexa y no convexa y teoría dual
   2.2. Puesta a prueba de los conocimientos adquiridos: problemas teóricos
3. Optimización convexa y el problema de control óptimo
   3.1. Estudio de la conexión entre el problema de control óptimo y su formulación como un problema de programación lineal
4. Enfoque primal-dual del problema de control óptimo: formulación general para problemas de pequeña escala
   4.1. Teoría dual aplicada al problema de control óptimo formulado como un programa lineal
   4.2. Formulación del problema de control óptimo como un problema de saddle point (o de punto de silla) del Lagrangiano
   4.3. Estudio de métodos de gradiente para resolver el problema de saddle point del Lagrangiano: Arrow-Hurwicz y dual-ascent
   4.4. Implementación de un nuevo algoritmo de aprendizaje por refuerzo basado en el problema de saddle point del Lagrangiano, resuelto mediante dual-ascent.
      4.4.1. Versión con conocimiento del modelo
      4.4.2. Versión sin conocimiento del modelo: aproximación estocástica del gradiente
   4.5. Pruebas de validación del algoritmo desarrollado
   4.6. Comparación con otros algoritmos ampliamente utilizados: SARSA y Q-learning
5. Enfoque primal-dual del problema de control óptimo: formulación general para problemas en los que el espacio de estados es muy grande o continuo
   5.1. Estudio de la problemática que surge cuando el espacio de acciones es muy grande o continuo
   5.2. Métodos de resolución del problema de control óptimo a través de aproximaciones lineales de la función valor
      5.2.1. Estudio de algoritmos que emplean actualmente este tipo de aproximaciones
      5.2.2. Adaptación del algoritmo desarrollado a esta nueva situación en la que la función valor se aproxima de forma lineal
      5.2.3. Pruebas de validación del algoritmo desarrollado

5.2.4. Comparación con otro algoritmo ampliamente utilizado: LSPI

5.3. Desarrollo de una demostración teórica sólida que garantice poder emplear cualquier tipo de función (lineal o no) para aproximar la función valor

5.4. Métodos de resolución del problema de control óptimo a través de aproximaciones no lineales de la función valor, basadas en redes neuronales

5.4.1. Formación teórica en redes neuronales

5.4.2. Aprendizaje de implementación de redes neuronales en Python a través de librerías ampliamente usadas en el mundo de la industria: TensorFlow y Keras

5.4.3. Unión de los conceptos relativos al aprendizaje por refuerzo y a redes neuronales: deep reinforcement learning o aprendizaje por refuerzo profundo

5.4.4. Estudio de las técnicas y algoritmos más empleados a día de hoy para este tipo de problemas: Neural Fitted Q-iteration y Deep Q-learning networks

5.4.5. Adaptación del algoritmo desarrollado a esta nueva situación en la que la función valor se aproxima mediante redes neuronales

5.4.6. Pruebas de validación del algoritmo desarrollado

Método de aprendizaje: libros, artículos, video-cursos y ejemplos de implementación real.

Herramientas, lenguajes y librerías empleadas para la implementación de los algoritmos y simulación de los problemas: Matlab, Python, TensorFlow y Keras.

9. Análisis de las condiciones de trabajo vividas en el centro, así como una reflexión sobre las características del entorno laboral y de la organización en la que han tenido lugar las prácticas.

Considero que las condiciones de trabajo que he tenido en el Grupo de Aplicaciones de Procesado de Señales han sido muy satisfactorias. Han supuesto un compromiso adecuado entre (a) tiempo invertido, (b) satisfacción personal conseguida al aprender sobre una temática (a juicio personal) tan interesante, (c) ampliación de conocimientos y formación en campos muy novedosos y (d) salario percibido.

En lo que se refiere al entorno laboral tampoco tengo ninguna queja. Estaba rodeado de un equipo pequeño de gente enormemente motivada con el trabajo que hace, lo cual favorece también a la motivación personal, y en consecuencia a alcanzar los objetivos fijados. En resumen, el entorno ha sido de lo más “amigable” posible: he podido consultar al resto de miembros del grupo, y especialmente a Sergio Valcarcel, cualquier duda que me surgiese y siempre he recibido buen trato.

Al margen de la línea principal de trabajo, cada jueves se han realizado reuniones grupales de seguimiento en las que todos de miembros del grupo que hemos comenzado recientemente la labor de investigación exponíamos nuestro trabajo y progreso. De este modo, he adquirido nociones más avanzadas sobre otras materias como son las redes neuronales, de gran utilidad en combinación con mi línea de investigación. Por tanto, a lo largo de las prácticas he podido nutrirme, no sólo del conocimiento que me proporcionaba mi trabajo, sino del generado por los demás.

Para concluir, creo que las prácticas que he realizado han tenido una organización correcta, pues ha habido un seguimiento continuo de mi trabajo, nunca me ha faltado apoyo y las condiciones de trabajo han sido adecuadas para el tipo de prácticas que son.