Filtrando eventos de seguridad en forma conservativa mediante deep learning

Leandro Ferrado Matías Cuenca-Acuna

Argentina Software Design Center (ASDC), Intel Security - Córdoba, Argentina
{leandro.ferrado,francisco.m.cuenca-acuna}@intel.com

5 de Septiembre del 2016

Contenidos:

Contenidos:

Resumen

- Aprendizaje Profundo

Resumen

- Aprendizaje Profundo
- Distribución del cómputo

Resumen

- Aprendizaje Profundo
- Distribución del cómputo
- Clasificación en señales de EEG

Contenidos:

Justificación

- Problemas complejos $\Rightarrow$ Aprendizaje profundo

Justificación

- Problemas complejos $\Rightarrow$ Aprendizaje profundo
- Costo computacional $\Rightarrow$ Sistema distribuido (local - clúster)

Justificación

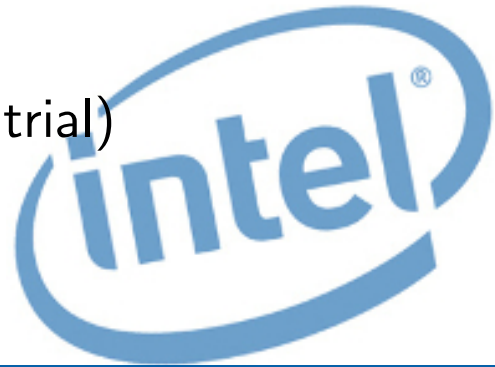
- Problemas complejos $\Rightarrow$ Aprendizaje profundo
- Costo computacional $\Rightarrow$ Sistema distribuido (local - clúster)
- Reutilizable por desarrolladores
  - Interfaz y documentación claras
  - Transparencia del cálculo paralelo involucrado

Justificación

- Problemas complejos $\Rightarrow$ Aprendizaje profundo
- Costo computacional $\Rightarrow$ Sistema distribuido (local - clúster)
- Reutilizable por desarrolladores
  - Interfaz y documentación claras
  - Transparencia del cálculo paralelo involucrado
- Uso de autocodificadores

Justificación

- Problemas complejos $\Rightarrow$ Aprendizaje profundo
- Costo computacional $\Rightarrow$ Sistema distribuido (local - clúster)
- Reutilizable por desarrolladores
  - Interfaz y documentación claras
  - Transparencia del cálculo paralelo involucrado
- Uso de autocodificadores
- Aplicación en problemática compleja:
  - "Clasificación sobre señales cerebrales"
  - Interfaz cerebro-computadora (ICC)
  - Comunicación confiable usando épocas únicas (single-trial)



Contenidos:

Objetivos generales

- Desarrollar un framework con algoritmos de aprendizaje profundo para entrenamiento y validación de redes neuronales, con posibilidad de distribuir el trabajo computacional sobre una computadora y/o una red de ellas.
- Aplicar la implementación obtenida en problemas de clasificación sobre datos de señales cerebrales, con el fin de analizar la potencialidad de las herramientas desarrolladas.

Objetivos específicos

- Definir funcionalidades y herramientas del framework.
- Investigar e implementar un motor de procesamiento distribuido escalable a clústeres.
- Lograr concurrencia para el entrenamiento de redes neuronales profundas a través de los nodos de un clúster.
- Obtener un software con código suficientemente documentado en todas sus funcionalidades.

Objetivos específicos

- Conseguir una interfaz para que abstraiga el cómputo paralelo implicado.
- Desarrollar un protocolo de experimentación sobre el caso de aplicación.
- Llevar adelante las pruebas especificadas.
- Obtener resultados mejores que el azar sobre los datos tratados.

Contenidos:

Alcance

1. Enfoque de autocodificadores para pre-entrenamiento de redes neuronales profundas.

Alcance

1. Enfoque de autocodificadores para pre-entrenamiento de redes neuronales profundas.
2. Software con estructura simple y código fácil de leer
   $\Rightarrow$ Lenguaje: Python

Alcance

1. Enfoque de autocodificadores para pre-entrenamiento de redes neuronales profundas.
2. Software con estructura simple y código fácil de leer
   $\Rightarrow$ Lenguaje: Python
3. Uso de motor de procesamiento distribuido Apache Spark™(libre)

Alcance

1. Enfoque de autocodificadores para pre-entrenamiento de redes neuronales profundas.
2. Software con estructura simple y código fácil de leer
   $\Rightarrow$ Lenguaje: Python
3. Uso de motor de procesamiento distribuido Apache Spark™(libre)
4. Distribución de cómputo a nivel local y nivel clúster

Alcance

1. Enfoque de autocodificadores para pre-entrenamiento de redes neuronales profundas.
2. Software con estructura simple y código fácil de leer
   $\Rightarrow$ Lenguaje: Python
3. Uso de motor de procesamiento distribuido Apache Spark^{TM}(libre)
4. Distribución de cómputo a nivel local y nivel clúster
5. Bases de datos adquiridas de forma gratuita

Alcance

1. Enfoque de autocodificadores para pre-entrenamiento de redes neuronales profundas.
2. Software con estructura simple y código fácil de leer
   $\Rightarrow$ Lenguaje: Python
3. Uso de motor de procesamiento distribuido Apache Spark^{TM}(libre)
4. Distribución de cómputo a nivel local y nivel clúster
5. Bases de datos adquiridas de forma gratuita
6. Señales de EEG ya pre-procesadas (no incluye investigación de fenómeno de estudio)

Contenidos:

Metodología

Ciclo de vida: Modelo en cascada (arquitectura estable y requisitos bastantes predefinidos)

Etapas

Metodología

Ciclo de vida: Modelo en cascada (arquitectura estable y requisitos bastantes predefinidos)

Etapas

1. Estudio estratégico y recolección de requisitos

Metodología

Ciclo de vida: Modelo en cascada (arquitectura estable y requisitos bastantes predefinidos)

Etapas

1. Estudio estratégico y recolección de requisitos
2. Diseño de propuesta de solución

Metodología

Ciclo de vida: Modelo en cascada (arquitectura estable y requisitos bastantes predefinidos)

Etapas

1. Estudio estratégico y recolección de requisitos
2. Diseño de propuesta de solución
3. Obtención de recursos y del marco teórico

Metodología

Ciclo de vida: Modelo en cascada (arquitectura estable y requisitos bastantes predefinidos)

Etapas

1. Estudio estratégico y recolección de requisitos
2. Diseño de propuesta de solución
3. Obtención de recursos y del marco teórico
4. Producción y testeo del software

Metodología

Ciclo de vida: Modelo en cascada (arquitectura estable y requisitos bastantes predefinidos)

Etapas

1. Estudio estratégico y recolección de requisitos
2. Diseño de propuesta de solución
3. Obtención de recursos y del marco teórico
4. Producción y testeo del software
5. Puesta en marcha y experimentación

Metodología

Ciclo de vida: Modelo en cascada (arquitectura estable y requisitos bastantes predefinidos)

Etapas

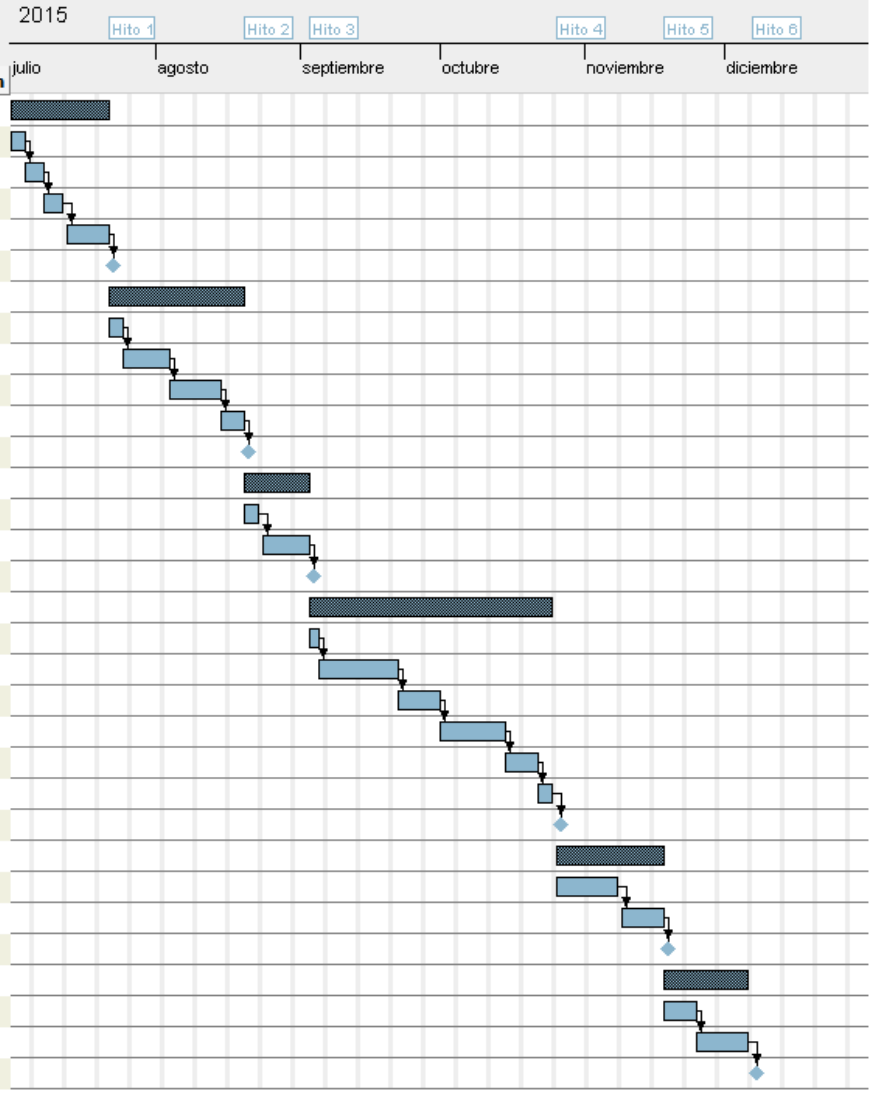
1. Estudio estratégico y recolección de requisitos
2. Diseño de propuesta de solución
3. Obtención de recursos y del marco teórico
4. Producción y testeo del software
5. Puesta en marcha y experimentación
6. Análisis de resultados e informe de trabajo

Contenidos:

Cronograma

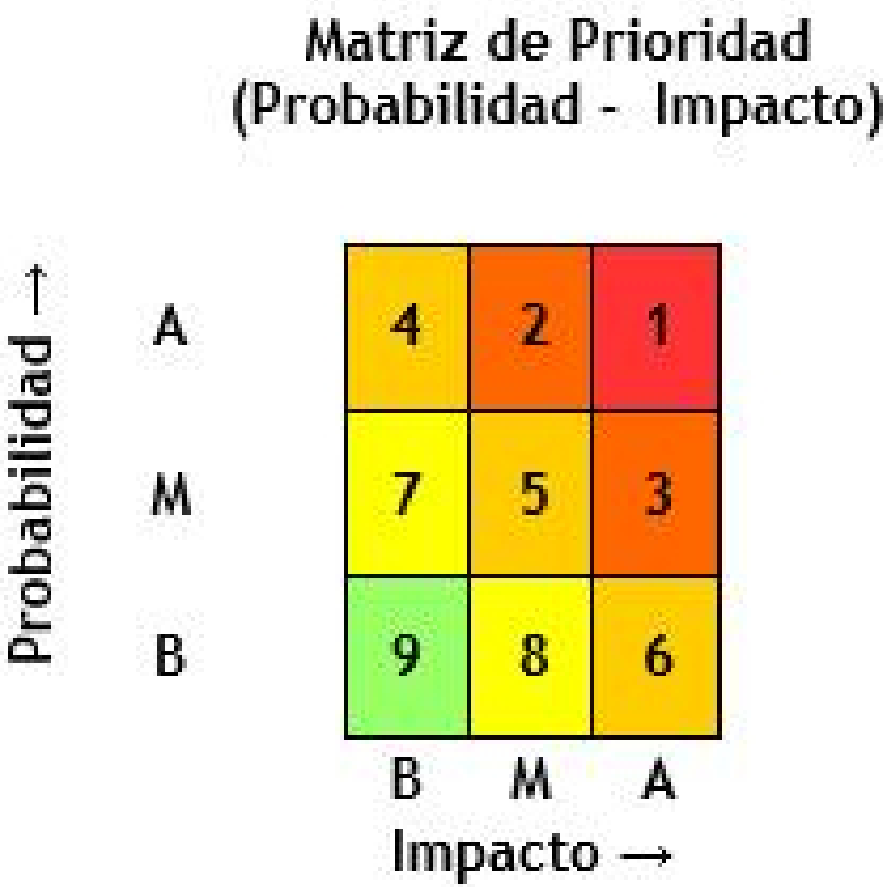
Cronograma de actividades

Nombre	Fecha de inicio	Fecha de fin
Etapa 1	1/07/15	21/07/15
Actividad 1.1	1/07/15	3/07/15
Actividad 1.2	4/07/15	7/07/15
Actividad 1.3	8/07/15	11/07/15
Actividad 1.4	13/07/15	21/07/15
Hito 1	22/07/15	22/07/15
Etapa 2	22/07/15	19/08/15
Actividad 2.1	22/07/15	24/07/15
Actividad 2.2	25/07/15	3/08/15
Actividad 2.3	4/08/15	14/08/15
Actividad 2.4	15/08/15	19/08/15
Hito 2	20/08/15	20/08/15
Etapa 3	20/08/15	2/09/15
Actividad 3.1	20/08/15	22/08/15
Actividad 3.2	24/08/15	2/09/15
Hito 3	3/09/15	3/09/15
Etapa 4	3/09/15	24/10/15
Actividad 4.1	3/09/15	4/09/15
Actividad 4.2	5/09/15	21/09/15
Actividad 4.3	22/09/15	30/09/15
Actividad 4.4	1/10/15	14/10/15
Actividad 4.5	15/10/15	21/10/15
Actividad 4.6	22/10/15	24/10/15
Hito 4	26/10/15	26/10/15
Etapa 5	26/10/15	17/11/15
Actividad 5.1	26/10/15	7/11/15
Actividad 5.2	9/11/15	17/11/15
Hito 5	18/11/15	18/11/15
Etapa 6	18/11/15	5/12/15
Actividad 6.1	18/11/15	24/11/15
Actividad 6.2	25/11/15	5/12/15
Hito 6	7/12/15	7/12/15



Contenidos:

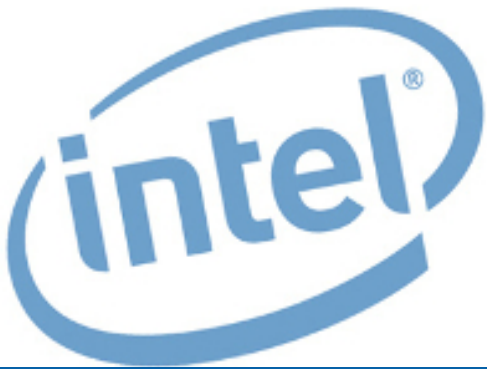
Riesgos



Estrategias de Respuesta:
Evitar (1-3)
Mitigar (4-6)
Aceptar (7-9)

Riesgos

R001 **Incapacidad de integración del motor de procesamiento distribuido en las actividades de producción del software.**
Prioridad: 6 $\Rightarrow$ Mitigar probabilidad



Riesgos

R001 **Incapacidad de integración del motor de procesamiento distribuido en las actividades de producción del software.**
Prioridad: 6 ⇒ Mitigar probabilidad

R002 **Incompatibilidades al migrar el trabajo del framework de la PC a los nodos del clúster.**
Prioridad: 3 ⇒ Evitar

Riesgos

R001 **Incapacidad de integración del motor de procesamiento distribuido en las actividades de producción del software.**
Prioridad: 6 ⇒ Mitigar probabilidad

R002 **Incompatibilidades al migrar el trabajo del framework de la PC a los nodos del clúster.**
Prioridad: 3 ⇒ Evitar

R003 **Carencia/deficiencia de la base de datos correspondiente al caso de aplicación elegido.**
Prioridad: 6 ⇒ Mitigar probabilidad

Riesgos

R001 **Incapacidad de integración del motor de procesamiento distribuido en las actividades de producción del software.**
Prioridad: 6 ⇒ Mitigar probabilidad

R002 **Incompatibilidades al migrar el trabajo del framework de la PC a los nodos del clúster.**
Prioridad: 3 ⇒ Evitar

R003 **Carencia/deficiencia de la base de datos correspondiente al caso de aplicación elegido.**
Prioridad: 6 ⇒ Mitigar probabilidad

R004 **Inconsistencias que surjan durante la experimentación, en el comportamiento de las funcionalidades elegidas.**
Prioridad: 6 ⇒ Mitigar impacto

Riesgos

R005 **Nodos de clústeres no disponibles por el tiempo necesario para completar la experimentación.**

Prioridad: 5 $\Rightarrow$ Mitigar impacto

Riesgos

R005 **Nodos de clústeres no disponibles por el tiempo necesario para completar la experimentación.**
Prioridad: 5 ⇒ Mitigar impacto

R006 **No disponibilidad del responsable de la ejecución del proyecto.**
Prioridad: 6 ⇒ Mitigar impacto

Riesgos

R005 **Nodos de clústeres no disponibles por el tiempo necesario para completar la experimentación.**
Prioridad: 5 ⇒ Mitigar impacto

R006 **No disponibilidad del responsable de la ejecución del proyecto.**
Prioridad: 6 ⇒ Mitigar impacto

R007 **No disponibilidad de los directores del proyecto.**
Prioridad: 8 ⇒ Aceptar

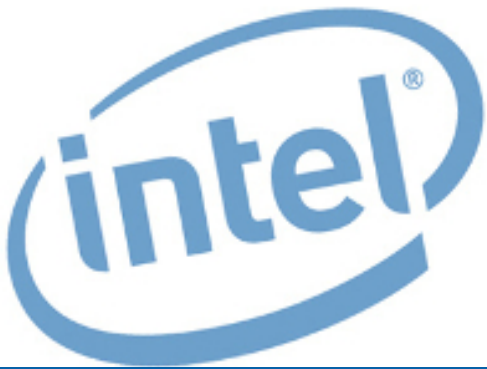
Riesgos

R005 **Nodos de clústeres no disponibles por el tiempo necesario para completar la experimentación.**
Prioridad: 5 ⇒ Mitigar impacto

R006 **No disponibilidad del responsable de la ejecución del proyecto.**
Prioridad: 6 ⇒ Mitigar impacto

R007 **No disponibilidad de los directores del proyecto.**
Prioridad: 8 ⇒ Aceptar

R008 **Experimentación lenta e insuficiente sobre el caso de aplicación.**
Prioridad: 5 ⇒ Mitigar impacto



Contenidos:

Recursos

Disponibles

- Computadora Personal (multi-núcleo)
- Bibliografía (FICH, MinCyT)
- Servicios del Proyecto (Internet, electricidad, etc.)

Recursos

Disponibles

- Computadora Personal (multi-núcleo)
- Bibliografía (FICH, MinCyT)
- Servicios del Proyecto (Internet, electricidad, etc.)

Necesarios

- Software (tecnologías libres)
- Bases de datos p/ caso de aplicación
- Servicio de clúster

Contenidos:

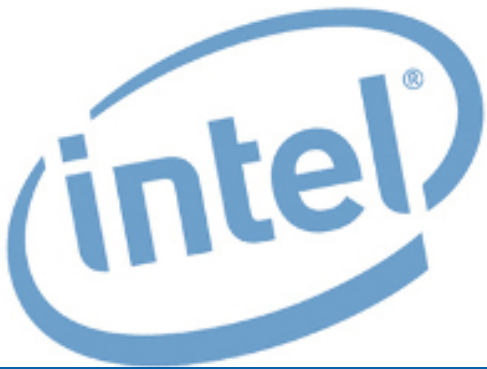
Costos

- Bienes de capital:
  - Notebook Asus N56VB S3065H DF Intel Core i5
    - Valor a nuevo (VN): $22000
    - Valor residual (VR): $1500
    - Vida útil (VU): 12000 horas
    - Monto de amortización: $\frac{VN-VR}{VU} \cdot$ horas de uso $\approx$ **$697**

Costos

- Bienes de capital:
  - Notebook Asus N56VB S3065H DF Intel Core i5
    - Valor a nuevo (VN): $22000
    - Valor residual (VR): $1500
    - Vida útil (VU): 12000 horas
    - Monto de amortización: $\frac{VN-VR}{VU} \cdot$ horas de uso $\approx$ **$697**
- Consultorías y Servicios:
  - Servicio de clúster computacional (Costos directos):
    - 4 nodos por clúster
    - 4 núcleos por nodo
    - 7GB de RAM por nodo
    - 100GB de almacenamiento

      Costo: $3,60/h **Total $864.**



Costos

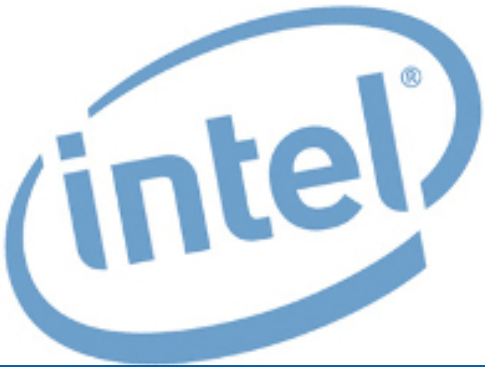
- Recursos humanos (Costos directos):
  - Remuneración propia:
    - Rol de analista funcional (Etapas 1, 2, 3 y 6).
      Monto: $160/h. Total: **$34080.**
    - Rol de analista programador y tester (Etapas 4 y 5).
      Monto: $130/h. Total: **$25350.**
    - **Total: $59430.**

Costos

- Recursos humanos (Costos directos):
  - Remuneración propia:
    - Rol de analista funcional (Etapas 1, 2, 3 y 6).
      Monto: $160/h. Total: **$34080.**
    - Rol de analista programador y tester (Etapas 4 y 5).
      Monto: $130/h. Total: **$25350.**
    - **Total: $59430.**
  - Remuneración del Director de proyecto:
    - Rol de Líder de Proyecto. 120hs.
      Monto: $300/h. Total: **$36000.**

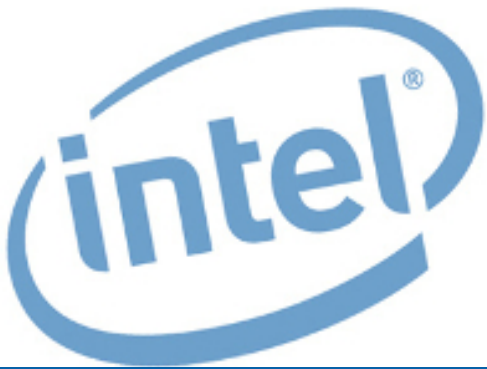
Costos

- Recursos humanos (Costos directos):
  - Remuneración propia:
    - Rol de analista funcional (Etapas 1, 2, 3 y 6).
      Monto: $160/h. Total: **$34080.**
    - Rol de analista programador y tester (Etapas 4 y 5).
      Monto: $130/h. Total: **$25350.**
    - **Total: $59430.**
  - Remuneración del Director de proyecto:
    - Rol de Líder de Proyecto. 120hs.
      Monto: $300/h. Total: **$36000.**
  - Remuneración del Co-director de proyecto:
    - Rol de Líder de Proyecto. 88hs.
      Monto: $300/h. Total: **$26400.**



Costos

- Materiales e Insumos (Costos directos):
  - Librería. Total: **$620.**



Costos

- Materiales e Insumos (Costos directos):
  - Librería. Total: **$620.**
- Viajes y Viáticos (Costos directos):
  - Transporte urbano. Total: **$1224.**
  - Merienda. Total: **$250.**

Costos

- Materiales e Insumos (Costos directos):
  - Librería. Total: **$620.**
- Viajes y Viáticos (Costos directos):
  - Transporte urbano. Total: **$1224.**
  - Merienda. Total: **$250.**
- Otros costos (Costos indirectos):
  - Electricidad. Total: **$234.**
  - Conexión a Internet. Total: **$350.**

Costos

- Materiales e Insumos (Costos directos):
  - Librería. Total: **$620.**
- Viajes y Viáticos (Costos directos):
  - Transporte urbano. Total: **$1224.**
  - Merienda. Total: **$250.**
- Otros costos (Costos indirectos):
  - Electricidad. Total: **$234.**
  - Conexión a Internet. Total: **$350.**
- Presupuesto total: **$126.069**

Bibliografía:

- Karau, H. et. al. (2015): "Learning Spark. Lightning-Fast Big Data Analysis". O'Reilly Media.
- Pentreath, N. (2015): "Machine Learning with Spark". Packt Publishing.
- Langtangen, H. (2008): "Python Scripting for Computational Science". Springer. Tercera edición.
- González-Castañeda, E. F., Torres-García, A. A. et al (2014): "Sonificación de EEG para la clasificación de palabras no pronunciadas" en Research in Computing Science 74.
- DaSalla, C. et al (2009): "Single-trial classification of vowel speech imagery using common spatial patterns" en Neural Networks 22 (9) (pp. 1334-1339).

Bibliografía:

- I. E. Gareis, G. Gentiletti, R. C. Acevedo, H. L. Rufiner (2011): "Feature extraction on Brain Computer Interfaces using Discrete Dyadic Wavelet Transform: Preliminary results" en Journal of Physics: Conference Series (IOP), Volume 313, Number 12011 (pp. 1-7).
- Erhan, D. et al (2010): "Why does unsupervised pre-training help deep learning?" en J. Mach. Learn. Res., 11:625-660.
- Bengio, Y (2009): "Learning deep architectures for AI" en Foundations and Trends®in Machine Learning archive.

Bibliografía:

- Haykin, S. (2009): "Neural Networks and Learning Machines". Prentice Hall. Tercera edición.
- Bishop, C.M. (1995): "Neural Networks for Pattern Recognition". Oxford: Oxford University Press.
- Project management institute, inc (2013): "Guía de los Fundamentos para la Dirección de Proyectos". PMI Book. Cuarta edición.
- Emiliani, F. (1995): "Proyectos de investigación científica". UNL-CONICET-ACNL.
- Ander Egg, E, Aguilar Idañez, M. (1998): "Cómo elaborar un proyecto". Editorial Lumen.