

STUDY OF CRITICAL VITAL SIGNS USING DEEP LEARNING

Diego Felipe Rodriguez Chaparro (diefrodriguezcha@unal.edu.co)

Ingenieria de Sistemas y Computacion, Facultad de Ingenieria

Universidad Nacional de Colombia, 2017

ABSTRACT

Con la creciente popularidad de Deep Learning en el campo de la ciencia, es inevitable experimentar y descubrir el alcance de esta poderosa herramienta y todo lo que puede ofrecer. En este trabajo se explora la posibilidad de hacer uso de las metodologías de Deep Learning en el campo de la medicina, específicamente orientado al estudio de signos vitales de los pacientes ingresados a Urgencias que están en un estado crítico. Haciendo uso de un data set de dominio público del Massachusetts General Hospital junto a los módulos de aprendizaje de Python, se quiere hacer uso de Deep Learning para que dados los signos vitales del paciente, el programa pueda calcular la probabilidad de supervivencia de este.

Palabras Clave - Deep Learning, Emergency Room, vital signs, Python, Keras

INTRODUCCIÓN

En la actualidad, el uso del aprendizaje de máquina (“Machine Learning”) se ha popularizado a través de múltiples campos, especialmente aquellos que manejan grandes volúmenes de información, para la automatización de tareas y modelamiento de patrones. Desde la década de 1970 los investigadores han estado trabajando en el estudio, creación y refinamiento de técnicas que le permitan a una computadora aprender a partir de datos observacionales, y de estos esfuerzos surgieron las redes neuronales. Una red neuronal estándar consiste de varios procesos conexos llamados neuronas, los cuales generan activaciones específicas, dependiendo de los estímulos que reciban, como están conectadas entre ellas o las acciones que estos estímulos generan [1].

Conforme fueron pasando los años y las técnicas se refinaron más, en los últimos tiempos se popularizó el uso del aprendizaje profundo (“Deep Learning”) por el alcance que ofrece este paradigma de la computación. Sus aplicaciones van desde el campo de la publicidad con el análisis de nuestros patrones de búsqueda para determinar el tipo de publicidad que probablemente llamaría más nuestra atención hasta el campo de la física, con el estudio de grandes cantidades de datos en brutos para hacer predicciones de futuros eventos cósmicos. Sus aplicaciones parecen ilimitadas.

Pero el campo que goza de las aplicaciones más atractivas es seguramente el de la medicina, con el uso de estas metodologías de análisis, los doctores son capaces de obtener diagnósticos cada vez más precisos y así prevenir enfermedades en etapas más tempranas. En la sala de emergencias donde cada segundo cuenta para salvar vidas, no solo es importante el rápido transporte y atención a la víctima, también dar una evaluación rápida y diagnóstico correcto es vital para maximizar la supervivencia del paciente [2]. El propósito de este trabajo será determinar la viabilidad de aplicar esta metodología de Deep Learning para analizar los signos vitales del paciente en condición crítica y de acuerdo a estos, junto al reporte de intervenciones en la Sala de Emergencias, determinar la probabilidad de supervivencia del paciente.

TRABAJOS RELACIONADOS

Como se menciono anteriormente, en los últimos años el tema del aprendizaje profundo ha ido creciendo en popularidad debido a sus aplicaciones multidisciplinarias, aplicaciones que parecen no tener fin. El uso de este modelo ha aportado a la construcción de diversos campos como los son las redes neuronales, modelos gráficos, aprendizaje de características, aprendizaje no supervisado, reconocimiento de patrones, optimización y el procesamiento de señales [3]. En el procesamiento de imágenes y su clasificación es donde está nuestro enfoque, específicamente el campo médico con el análisis de imágenes biomédicas, desde el año 1865 con el descubrimiento de los rayos X y su posterior uso se logró un mejor entendimiento de cómo se manifestaban ciertas enfermedades dentro del cuerpo [4], esta tecnología abrió las puertas para un nuevo mundo de desarrollo del cual todos podemos gozar hoy en día.

La mezcla de estas dos disciplinas (Deep Learning y Medicina) ha permitido entregar a los expertos diagnósticos más precisos, permitiendo salvar vidas. Aunque muestra grandes avances, la aplicación de estas tecnologías sigue en etapas tempranas y no está exenta de errores, en el trabajo de B.Gerazov y R.Conceicao se puede evidenciar esto; en el estudio de mamografías por rayos X se presenta el problema que el tejido adiposo natural del seno exhibe un contraste dieléctrico muy similar al de los tumores malignos, haciendo que su detección tenga una alta tasa de error [5], el objetivo de este trabajo era mejorar el porcentaje de clasificación de tumores a través del Deep Learning, lo cual lograron utilizando un acercamiento híbrido: Utilizaron una Deep Neuronal Network, a la cual no solo le dieron como entrada

imagenes de mamografias, tambien le dieron los resultados de clasificacion de implementaciones convencionales de Machine Learning.

Con el trabajo anterior se lograron resultados prometedores teniendo en cuenta las limitantes existentes en cuanto al volumen reducido de la informacion disponible. Ese es solo uno de los aspectos que afecta este problema tan complejo; siendo este una tarea de clasificacion se debe aspirar a lograr un punto de equilibrio optimo entre la precision de las clasificaciones obtenidas y el costo computacional [6]; este problema puede ser abordado de dos maneras diferentes ya que el data set provee dos tipos de informacion: Provee lecturas visuales de los signos vitales del paciente que muestra el estado de los varios sistemas que componen el cuerpo; siendo estas señales tan complejas, es necesario entrenar al sistema para que reconozca la situacion actual del paciente a traves de estas imagenes. En el trabajo de D. Yudin y B. Zeno [7], utilizando dos redes neuronales convolucionales (CNN) previamente entrenadas, la maquina logro una tasa de éxito del 83.2% al tratar de clasificar imagenes en 10 categorias especiales, que eran “carrera de autos”, “ceremonia”, “football”, “reunion”, “demostracion”, “picnic”, “tenis”, “concierto”, “trafico” y “natacion”.

El otro tipo de informacion que el dataset provee es una bitacora que contiene los registros de los procedimientos realizados para estabilizar al paciente. En trabajos anteriores, donde se ha buscado predecir futuros resultados estudiando solo el presente que varia, se ha propuesto utilizar un analisis estadistico de los datos y producir asi tasas de éxito o de fracaso [8]. Una probabilidad que hasta ahora se esta explorando.

El inconveniente mas grande que surge en estos dos enfoques, es la cantidad de datos disponibles para el entrenamiento y posteriormente para las prueba (especialmente para el entrenamiento). Los proyectos medicos que utilizan deep learning se basan en el reconocimiento de patrones e indicios y para mejorar la efectividad del proyecto es necesario tener datos suficientes como se dijo anteriormente. Una solucion para el reconocimiento visual es la de generar muestras de datos sinteticos que se encuentren entre parametros aceptables y sirvan para mejorar el entrenamiento de la red neuronal [9], los resultados de este estudio apoyaron la hipotesis de los investigadores, las maquinas entrenadas con estas muestras sinteticas mostraron un mejor desempeño que las otras entrenadas con metodos tradicionales; aplicado a este proyecto, la escases de datos no representaria un obstaculo al mejorar su eficacia en la clasificacion; siendo una opcion viable y tal vez, menos costosa, no se debe evitar datos del mundo real, podrian presentarse “imbalances de datos” al momento de catalogar anomalias [10].

METODOLOGIA

Como herramientas principales para la realizacion de este proyecto, se uso la librería Keras para python y el data set “MGH/MF waveform database” de la pagina physio.net [11], este data set contiene 250 registros de pacientes ingresados a la Sala de Emergencias del Massachusett’s General Hospital, donde se tiene toda la informacion del estado del paciente junto a una digitalizacion de las señales de los signos vitales de este.

En cada uno de los registros de esta base de datos se encuentra la siguiente informacion del paciente:

Datos del Paciente

- Edad y genero
- Diagnostico/Cirugia
- Historia medica pertinente

Datos del Electrocardiograma

- Ritmo subyacente
- Disturbios en el ritmo
- Interpretacion del ECG

Datos Hemodinamicos

- Presion arterial
- Presion arterial pulmonar (PAP)
- Presion de la auricula derecha (RAP)

Patrones de Ondas (Breve descripcion)

Datos Respiratorios

- Ritmo
- Modo de ventilación

Algunos de estos registros incluyen una bitacora de los procedimientos realizados a los pacientes durante su estadia en la sala de emergencias.

Puesto que muchos de los datos son de tipo cualitativo, se inoraran los que no sean decisivos para el modelo, y se implementara un valor numerico para aquellos que cumplen una funcion descriptiva, dejando la lista de datos utilizados asi:

Datos del Paciente

- Edad : Valor numerico de los años del paciente.
- Genero : 0 si es hombre, 1 si es femenino.
- Historia Medica Pertinente: 1 si el paciente tiene, 0 si no es asi.

Datos Hemodinamicos

- Presion Arterial : Valor numerico.
- Presion arterial pulmonar: Valor numerico
- Presion de la auricula derecha: Valor numerico

Datos Respiratorios

- Ritmo: Valor numerico
- Modo de ventilacion: Tendra un valor de 0 a 12 , un numero unico por cada uno de los valores distintos encontrados en esta descripcion.

Todos esta informacion son las entradas de datos para el modelo de deep learning, lo que haremos es poner estos datos de manera organizada en un archivo .csv para que el modelo los acepte sin problemas,

Dado que se cuenta con un data set pequeño, entrenaremos el modelo dividiendo los 250 registros en lotes, y uno tras otro pasaran por el modelo para optimizarlo y ajustarlo, a esto se le conoce como Gradiente Descendiente Estocastico, este enfoque ha

mostrado ser muy útil cuando se cuenta con información limitada, como es en este caso.

El modelo a usar obedecerá una disposición similar a la mostrada en la figura 1. Será un modelo con conexiones densas, que indica que cada nodo en una capa está conectado a todos los nodos de la capa siguiente, un acercamiento recomendado para tener mejores resultados. En la capa de salida, siendo este un problema de clasificación, sobre si el paciente vive o muere, al final en la capa de salida (output layer) van a ir dos nodos en vez de uno, que señalan los posibles resultados que se pueden obtener.

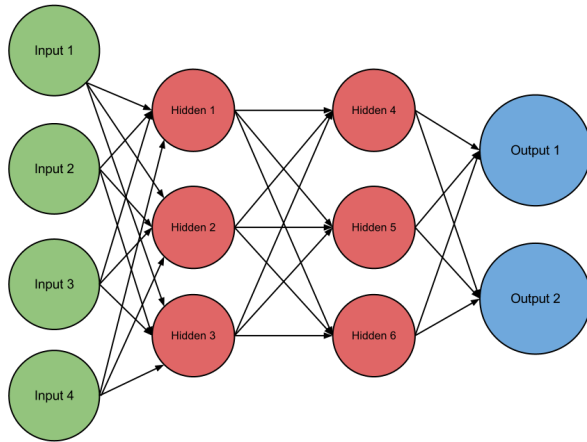


figura 1: Modelo secuencial

A medida que se avance en los experimentos, se variarán el número de capas ocultas y los nodos en estas para minimizar el valor del error y obtener predicciones más certeras.

La función de activación a utilizar en las capas ocultas va a ser la función ReLU (Rectified Linear Function) que viene incluida con la librería Keras para capturar comportamientos no lineales, y en la capa de salida usaremos la función “softmax”, que en problemas de clasificación como este, es la más recomendada.

En la parte de ajuste y optimización, para hacer la propagación reversa, utilizaremos la función de optimización ‘adam’ y la función de pérdida ‘categorical_crossentropy’, ambas vienen con Keras y son la elección recomendada para este tipo de problemas; para hacer más fácil la interpretación del modelo en términos de progreso de entrenamiento, añadiremos a los argumentos de la parte de compilación, que las métricas se den en términos de precisión, esto nos dirá la tasa de éxito lograda al clasificar las entradas en su respectiva categoría.

Y la parte de “fitting” del modelo se realizará con los resultados arrojados por el modelo y los resultados reales, estos se obtendrán interpretando las bitácoras de procedimientos de los registros para saber si el paciente vivió o murió luego de su estadía en la Sala de Emergencias.

DISEÑO E IMPLEMENTACIÓN

El enfoque del proyecto es probar la viabilidad de que un sistema de Deep Learning pueda predecir la capacidad de supervivencia de un individuo que es internado en la Sala de Emergencias de un hospital, siendo de un énfasis más experimental se decidió trabajar con la librería Keras de Python que permite de una manera ágil, implementar la solución y probarla; es necesario tener una versión de Python que esté entre la 2.7 y la 3.6, y tener una librería de aprendizaje inteligente que puede ser Tensorflow, CNTK o Theano. En este experimento se usará la versión r1.4 de Tensorflow para Linux junto a Python 2.7.

El modelo inicial tiene una disposición simple, se partió de un modelo de 3 capas ocultas sin incluir la capa de salida, las capas ocultas tendrán 50 nodos y la de resultados tendrá 2, que son los posibles resultados de esta clasificación binaria, después de este punto se variarán primero la cantidad de nodos por capa y luego el número de capas del modelo con el objetivo de reducir el valor de la función de pérdida y aumentar la tasa de éxito de las predicciones.

Como fue mencionado anteriormente, dado el número limitado de registros en el data set, se eligió entrenar el modelo siguiendo el método del Gradiente Estocástico Descendiente; luego de depurar el dataset y crear el nuevo archivo en formato .csv, de los 250 registros iniciales solo se cuenta con 233 entradas utilizables, y muchas de estas no cuentan con todos sus datos completos. Las razones de esto son descritas en los mismos registros, varios de ellos fueron perdidos por fallas del sistema o la situación que describe este era de una naturaleza tan extrema que no hubo tiempo para tomar esos registros.

Los registros se dividirán en lotes (batch) de 50 registros para la sección de “fitting” del modelo, también se hará uso inicialmente de 10 épocas para cada fase de entrenamiento y se tendrá un atributo de paciencia de valor 2, el cual señala el número de épocas que pueden pasar sin mejoría del modelo antes de detener el entrenamiento; una vez detenido se reformularán los valores anteriormente enunciados en aras de mejorar el modelo.

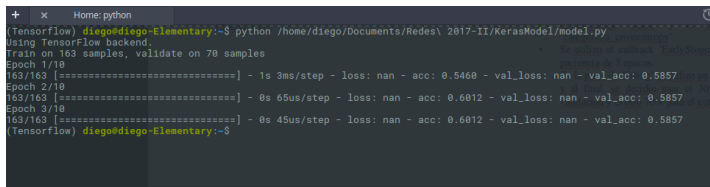
La ejecución y muestra de resultados se hará a través de la terminal, el formato de estos resultados son dados por la librería Keras, la cual permite un seguimiento claro del estado del modelo.

EVALUACIÓN DE RESULTADOS

Luego de varias pruebas con el modelo y de realizar los ajustes acordados, se logró el mejor resultado con la siguiente disposición:

- Modelo con 4 capas ocultas, cada una con 60 nodos, todos con la función de activación ‘ReLU’.
- La capa de salida con 2 nodos y activación ‘softmax’.
- En la parte de compilación, se usó la función optimizadora ‘adam’ y la función de pérdida ‘categorical_crossentropy’.
- Se utilizó el callback ‘EarlyStopping’ con un valor de paciencia de 3 épocas.
- En la parte de ‘fitting’ se utilizó un tamaño de lote de 64, y al final, se decidió usar el 30% de los datos para validación y el otro 70% para el entrenamiento.

Con esta configuracion se logro el siguiente resultado:



```
Home:python
(TensorFlow) diegodediego-Elementary:~$ python /home/diego/Documents/Redes/2017-II/KerasModel/model.py
Using TensorFlow backend.
Train on 163 samples, validate on 70 samples
Epoch 1/10
163/163 [=====] - 1s 3ms/step - loss: nan - acc: 0.5460 - val_loss: nan - val_acc: 0.5857
Epoch 2/10
163/163 [=====] - 0s 65us/step - loss: nan - acc: 0.6012 - val_loss: nan - val_acc: 0.5857
Epoch 3/10
163/163 [=====] - 0s 45us/step - loss: nan - acc: 0.6012 - val_loss: nan - val_acc: 0.5857
(TensorFlow) diegodediego-Elementary:~$
```

figura 2: Construcción del modelo

Se logro una precision del 60,12% en la capacidad de prediccion, lo cual es un buen resultado que respalda aun mas la idea de que el deep learning es una herramienta poderosa en este campo, especialmente en la tarea de clasificación.

Ocurrieron varias cosas inesperadas, luego de 3 epocas, el modelo se estanco y dejo de mejorar, sin importar la configuracion de este, y en todas las pruebas el modelo no arrojo un valor en la funcion de perdida, como se ve en la figura 2., estas fallas pueden ser atribuidas al estado del dataset, lo cual ilustra el papel tan importante que juega este elemento en el ejercicio de aprendizaje de maquina.

DISCUSIÓN DE RESULTADOS

Al principio los resultados obtenidos parecen prometedores con un acierto de mas del 50%, pero si los observamos detenidamente podemos ver que existen muchas fallas. El dataset presentaba varios datos faltantes, lo cual afecto en gran medida el valor de la funcion de perdida, el cual se ve en la figura 2, aparece con un valor 'nan', esto es conocido como "Problema de gradientes explosivos" que ocurre por una de dos razones, la primera es cuando el ritmo de aprendizaje en la funcion optimizadora es muy grande y los resultados de esa operación no son los correctos y el error se propaga, en este experimento ese no es el caso puesto que se uso la funcion 'adam' que calcula el mejor ritmo para evitar este problema. La segunda razon se da cuando la funcion de perdida tiene o valores muy altos o bajos y el programa se percata de la naturaleza anormal de estos valores, asi que le da a la funcion un valor de 'undefined' o 'nan', en este caso ocurrio por la falta de varios datos al entrenar el modelo.

Otro aspecto que influyo en gran parte fue que algunos de los datos involucrados eran cualitativos y no se contaba con una escala cuantitativa lo suficientemente acertada para sopesar estos argumentos de manera apropiada, sin el consejo de un medico experto los valores numericos que reemplazaron los valores culitativos no fueron del todo acertados. La razon de solo usar valores numericos es porque la librería Keras funciona en conjunto con Numpy, que solo acepta datos numericos para trabajar.

Tampoco se usaron datos de gran importancia como el Diagnostico y Cirugia por la razon expuesta anteriormente, sin tener una escala que mida correctamente la dificultad de un procedimiento medico no es posible usar estos datos que son uno de los factores determinantes para la supervivencia del paciente, de esta manera arrojando resultados incorrectos.

El hecho de que el modelo no haya tenido la curva de aprendizaje esperada y que se haya atascado luego de 3 epocas, pone en

evidencia la importancia de contar con un dataset de gran tamaño para tener suficientes registros para entrenar, validar y probar el modelo . El modelo fue configurado para actuar con esta limitante y como se vio en el resultado obtenido, se comporto de la mejor manera posible, teniendo el reducido numero de datos.

CONCLUSIONES

El uso Deep Learning en el campo de la medicina ha probado ser una herramienta con un gran potencial que promete ayudar a mejorar la condicion del ser humano, este experimento muestra solo un pequeño asomo del potencial que puede lograrse con el estudio de los signos vitales con la ayuda del aprendizaje de maquina.

En futuros experimentos de naturaleza multi-disciplinar como este es esencial trabajar con expertos en el tema que asesoren sobre los aspectos importantes del campo. Tambien se debe explorar la posibilidad de usra con otra librería diferente a Keras para trabajar con datos no numericos, puesto que se evidencio la dificultad de trabajar datos cualitativos como si fueran cuantitativos, no siempre se podra hallar o diseñar una escala numerica que ilustre estos aspectos de una manera acertada.

Se pueden mejorar los resultados obtenidos de varias maneras, entre ellas, contar con un dataset cuyos datos esten completos en su totalidad si es posible, dependiendo del objetivo del trabajo, saber elegir las funciones de optimizacion, perdida y activacion adecuados para los datos a estudiar. Haciendo uso de esto, es seguro que en futuras investigaciones se logran mejores resultados para el estudio de este campo de la medicina.

REFERENCIAS

- [1] Jürgen Schmidhuber, Deep learning in neural networks: An overview, In Neural Networks, Volume 61, 2015, Pages 85-117, ISSN 0893-6080, <https://doi.org/10.1016/j.neunet.2014.09.003>. (<http://www.sciencedirect.com/science/article/pii/S0893608014002135>)
- [2] Gordon L. Webb, Norman E. McSwain, Watts R. Webb, Charles Rodriguez, Emergency department deaths, In The American Journal of Surgery, Volume 159, Issue 4, 1990, Pages 377-379, ISSN 0002-9610, [https://doi.org/10.1016/S0002-9610\(05\)81275-9](https://doi.org/10.1016/S0002-9610(05)81275-9). (<http://www.sciencedirect.com/science/article/pii/S0002961005812759>)
- [3] S. B., L. D., H. L., H. L., & R. S. (2013, August). Guest Editors' Introduction: Special Section on Learning Deep Architectures. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 35(8).
- [4] W. B. (n.d.). History of Medical Imaging. In Proceedings of the American Philosophical Society(3rd ed., Vol. 152, pp. 349-361).
- [5] B. G., & R. C. (2017). Deep Learning for Tumour Classification in Homogeneous Breast Tissue in Medical Microwave Imaging. IEEE EUROCON 2017.
- [6] Affonso, C., Rossi, A. L. D., Vieira, F. H. A., & de Carvalho, A. C. P. D. L. F. (2017). Deep learning for biological

image classification. *Expert Systems with Applications*, 85, 114-122. doi:10.1016/j.eswa.2017.05.039

[7] Yudin, D., & Zeno, B. (2018). Event recognition on images by fine-tuning of deep neural networks doi:10.1007/978-3-319-68321-8_49

[8] Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 83, 187-205. doi:10.1016/j.eswa.2017.04.030

[9] A. U., S. P., E. E., & V. C. (2015, September). SYNTHETIC STRUCTURAL MAGNETIC RESONANCE IMAGE GENERATOR IMPROVES DEEP LEARNING PREDICTION OF SCHIZOPHRENIA. 2015 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING.

[10] M. C., X. S., Y. Z., D. W., & M. G. (2017). Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data.

[11] (n.d.). Retrieved November 25, 2017, from <https://physionet.org/physiobank/database/mghdb/>