



Aprende Machine Learning

antes de que sea demasiado tarde

GENERAL

Aprendizaje Profundo: una Guía rápida

🕒 noviembre 14, 2017 by Na8

Explicando Deep Learning y Redes Neuronales -sin código-

Intentaré explicar brevemente en qué consiste el **Deep Learning** ó *Aprendizaje Profundo* utilizado en Machine Learning describiendo sus componentes básicos.

Conocimientos Previos

Daré por sentado que el lector ya conoce [la definición de Machine Learning](#) y sus principales [aplicaciones en el mundo real](#) y el panorama de [algoritmos utilizados con mayor frecuencia](#). Nos centraremos en Aprendizaje Profundo aplicando [Redes Neuronales Artificiales](#).

Entonces, ¿cómo funciona el Deep Learning? Mejor un Ejemplo

El Aprendizaje Profundo es un método del Machine Learning que nos permite entrenar una Inteligencia Artificial para obtener una predicción dado un conjunto de entradas. Esta inteligencia logrará un nivel de cognición por jerarquías. Se puede utilizar [Aprendizaje Supervisado](#) o [No Supervisado](#).

Explicaré como funciona el Deep Learning mediante un ejemplo hipotético de predicción sobre quién ganará el próximo mundial de futbol. Utilizaremos aprendizaje supervisado mediante [algoritmos de Redes Neuronales Artificiales](#).

Para lograr las predicciones de los partidos de fútbol usaremos como ejemplo las siguientes entradas:

- Cantidad de Partidos Ganados
- Cantidad de Partidos Empatados
- Cantidad de Partidos Perdidos
- Cantidad de Goles a Favor

- Cantidad de Goles en Contra
- «Racha Ganadora» del equipo (cant. max de partidos ganados seguidos sobre el total jugado)

Y podríamos tener muchísimas entradas más, por ejemplo la puntuación media de los jugadores del equipo, o el score que da la FIFA al equipo. Como en cada partido tenemos a 2 rivales, deberemos estos 6 datos de entrada por cada equipo, es decir, 6 entradas del equipo 1 y otras 6 del equipo 2 dando un total de 12 entradas.

La predicción de salida será el resultado del partido: Local, Empate o Visitante.

Creamos una Red Neuronal

En la programación «tradicional» escribiríamos código en donde indicamos reglas por ejemplo «si goles de equipo 1 mayor a goles de equipo 2 entonces probabilidad de Local aumenta». Es decir que deberíamos programar artesanalmente unas reglas de inteligencia bastante extensa e inter-relacionar las 12 variables. Para evitar todo ese enredo y hacer que nuestro código sea escalaba y flexible a cambios recurrimos a las Redes Neuronales de Machine Learning para indicar una arquitectura de interconexiones y dejar que este modelo aprenda por sí mismo (y descubra él mismo relaciones de variables que nosotros desconocemos).

Lee aqui, sobre la [historia de las Redes Neuronales](#)

Vamos a crear una Red Neuronal con 12 valores de entrada (**Input Layer**) y con 3 neuronas de Salida (**Output Layer**). Las neuronas que tenemos en medio se llaman **Hidden Layers** y podemos tener muchas, cada una con una distinta cantidad de neuronas. Todas las neuronas estarán inter-conectadas unas con otras en las distintas capas como vemos en el dibujo. Las Neuronas son los círculos blancos.

- **La capa de entrada** recibe los datos de entrada y los pasa a la primer capa oculta.
- **Las capas ocultas** realizarán cálculos matemáticos con nuestras entradas. Uno de los desafíos al crear la Red Neuronal es decidir el número de capas ocultas y la cantidad de neuronas de cada capa.
- **La capa de Salida** devuelve la predicción realizada. En nuestro caso de 3 resultados discretos las salidas podrán ser «1 0 0» para Local, «0 1 0» para Empate y «0 0 1» para Visitante.

La cantidad total de capas en la cadena le da «profundidad» al modelo. De aquí es que surge la terminología de *Aprendizaje Profundo*.

¿Cómo se calcula la predicción?

Cada conexión de nuestra red neuronal está asociada a un **peso**. Este peso dictamina la importancia que tendrá esa relación en la neurona al multiplicarse por el valor de entrada. Los valores iniciales de peso se asignan aleatoriamente (SPOILER: más adelante los pesos se ajustarán solos).

Imitando a las neuronas biológicas, cada Neurona tiene una **Función de Activación**. Esta función determinará si la suma de sus valores recibidos (previamente multiplicados por el peso de la conexión) supera un umbral que hace que la neurona se active y dispare un valor hacia la siguiente capa conectada. Hay diversas Funciones de Activación conocidas que se suelen utilizar en estas redes.

Cuando todas las capas finalizan de realizar sus cálculos, se llegará a la capa final con una predicción. Por Ejemplo si nuestro modelo nos devuelve 0.6 0.25 0.15 está prediciendo que ganará Local con 60% probabilidades, será Empate 25% o que gane Visitante 15%.

Entrenando Nuestra Red Neuronal

Entrenar nuestra IA puede llegar a ser la parte más difícil del Deep Learning. Necesitamos:

1. Gran cantidad de valores en nuestro conjunto de Datos de Entrada
2. Gran poder de cálculo computacional

Para nuestro ejemplo de «predicción de Partidos de Futbol para el próximo Mundial» deberemos crear una base de datos con todos los resultados históricos de los Equipos de Fútbol en mundiales, en partidos amistosos, en clasificatorios, los goles, las rachas a lo largo de los años, etc.

Para entrenar nuestra máquina, deberemos alimentarla con nuestro conjunto de datos de entrada y comparar el resultado (local, empate, visitante) contra la predicción obtenida. Como nuestro modelo fue inicializado con pesos aleatorios, y aún está sin entrenar, las salidas obtenidas seguramente serán erróneas.

Una vez que tenemos nuestro conjunto de datos, comenzaremos un proceso iterativo: usaremos una función para comparar cuan bueno/malo fue nuestro resultado contra el resultado real. Esta función es llamada «**Función Coste**». **Idealmente queremos que**

nuestro coste sea cero, es decir sin error(cuando el valor de la predicción es igual al resultado real del partido). **A medida que entrena el modelo irá ajustando los pesos de inter-conexiones de las neuronas** de manera automática hasta obtener buenas predicciones. A ese proceso de «ir y venir» por las capas de neuronas se le conoce como Back-Propagation. Más detalle a continuación.

¿Cómo reducimos la función coste -y mejoramos las predicciones-?

Para poder ajustar los pesos de las conexiones entre neuronas haciendo que el coste se aproxime a cero usaremos una técnica llamada **Gradient Descent**. Esta técnica permite encontrar el mínimo de una función. En nuestro caso, buscaremos el mínimo en la Función Coste.

Funciona cambiando los pesos en pequeños incrementos luego de cada iteración del conjunto de datos. Al calcular la derivada (o gradiente) de la Función Coste en un cierto conjunto de pesos, podremos ver en que dirección «descender» hacia el mínimo global. Aquí se puede ver un ejemplo de Descenso de Gradiente en 2 dimensiones, imaginen la dificultad de tener que encontrar un mínimo global en 12 dimensiones!

Para minimizar la función de coste necesitaremos iterar por el conjunto de datos cientos de miles de veces (ó más), por eso es tan necesario **tener gran capacidad de cómputo** en el ordenador/nube en la que entrenamos la red.

La actualización del valor de los pesos se realizará automáticamente usando el **Descenso de Gradiente**. Esta es parte de la magia del Aprendizaje Profundo «Automático».

Una vez que finalizamos de entrenar nuestro Predictor de Partidos de Futbol del Mundial, sólo tendremos que alimentarlo con los partidos que se disputarán y podremos saber quién ganará Rusia 2018... Es un caso hipotético, pero es un ejercicio divertido para hacer.

¿Dónde puedo aprender más?

Hay muchos **tipos de Redes Neuronales**, **Convolutional Neural Networks** (CNN) usadas para Vision por Computadora o las **Recurrent Neural Networks** RNN para Procesamiento Natural del Lenguaje. Puedes leer mi artículo con **una sencilla implementación en Python con Keras**, un artículo con un **ejercicio en Python pero sin librerías**. O un artículo sobre la **historia de las Redes Neuronales**. Si quieres aprender el aspecto técnico del Aprendizaje Profundo puedo sugerir tomar un curso online. Actualmente el más popular es el de

Andrew Ng: [Deep Learning Specialization](#). Se puede cursar gratuitamente o pagar para obtener la certificación. O para un conocimiento general de Machine Learning también puedo recomendar [el curso estrella de Coursera](#). Un gran problema que surge al crear y entrenar Redes Neuronales es el [Overfitting al dar demasiada complejidad](#) a nuestra arquitectura de capas.

Si tienes dudas o preguntas sobre este tema puedo ayudarte, deja tus comentarios al finalizar este artículo.

Quieres preparar tu ambiente de desarrollo en Python?
[Sigue este tutorial!](#) Y crea tu primer [red neuronal aquí](#)

En Conclusión

- El Aprendizaje Profundo utiliza Algoritmos de Redes Neuronales Artificiales que imitan el comportamiento biológico del cerebro.
- Hay 3 tipos de Capas de Neuronas: de Entrada, Ocultas y de Salida.
- Las conexiones entre neuronas llevan asociadas un peso, que denota la importancia del valor de entrada en esa relación.
- Las neuronas aplican una Función de Activación para Estandarizar su valor de salida a la próxima capa de neuronas.
- Para entrenar una red neuronal necesitaremos un gran conjunto de datos.
- Iterar el conjunto de datos y comparar sus salidas producirá una Función Coste que indicará cuán alejado está nuestra predicción del valor real.
- Luego de cada iteración del conjunto de datos de entrada, se ajustarán los pesos de las neuronas utilizando el Descenso de Gradiente para reducir el valor de Coste y acercar las predicciones a las salidas reales.

Si te gusta este artículo puedes ayudarme difundiendo en Redes Sociales el enlace para que más personas lo encuentren.

Y si vas a crear tu propia máquina, recuerda seguir [los 7 pasos que comento en este artículo](#).

Para más actualizaciones te sugiero que te suscribas al blog o que me sigas en [Twitter!](#)

Nuevo! Ejercicio con [Redes Neuronales: Pronóstico de Ventas](#)

Entérate Primero – Suscripción

Anótate al blog y te notificará al publicarse el próximo post quincenal sobre Machine Learning. Súbete a la ola de Tecnología Disruptiva!

Email:

ENVIAR

Comparte el artículo:



Relacionado



¿Qué es Machine Learning? Una definición



Breve Historia de las Redes Neuronales Artificiales



Instalar ambiente de Desarrollo Python Anaconda para Aprendizaje Automático

Aprendizaje Automático

Aprendizaje Profundo

Big Data

cursos

Deep Learning

Definición

entrenar

Machine Learning

Modelos

predicción

Redes Neuronales

PRINCIPALES ALGORITMOS USADOS EN MACHINE LEARNING

REGRESIÓN LOGÍSTICA CON PYTHON PASO A PASO

7 comments



Luis · febrero 23, 2018

interesante articulo gracias por publicar. Por favor seria bueno contar con una practica de un ejemplo simple (con codigos incluidos) para tener la satisfaccion de habernos bautizado con algun modelo de ML ;). Personalmente estoy aprendiendo este tema y te agradezco por el Articulo. GRACIAS

Responder



Na8 · febrero 23, 2018

Hola Luis, muchas gracias por comentar en el Blog. Me alegro que te haya sido útil el artículo. Voy a seguir tu consejo y espero poder agregar un ejemplo con código dentro de poco. Saludos

Responder



oscar · junio 7, 2018

Te felicito sobre tu blog, hoy lo encuentre y lo estoy leyendo full ya que me intereso el tema y esta muy claro tus explicaciones.
Sobre el ejercicio ganará Rusia 2018, lo desarrollaste para que lo compartas. Saludos

Responder



Na8 · junio 18, 2018

Hola Oscar, gracias por las felicitaciones. Espero tener nuevos artículos sobre Machine Learning pronto. Sobre el ejercicio de Rusia no pude completarlo por falta de tiempo, pero te dejo un enlace en el que utilizaron Random Forest para hacer las predicciones. A ver que tan buena predicción tiene, pues daban una final Brasil-Alemania y por el momento uno empató y el otro perdió... es la pura prueba de que «la máquina» también puede fallar en sus predicciones... o de que el humano sigue siendo impredecible!. Saludos y espero que sigas visitando el blog.

- [Prediction of the FIFA World Cup 2018](#) (descargar el PDF en inglés)
- [Artículo que comenta las predicciones](#)

Responder



oscar · junio 18, 2018



muchas gracias por tu respuesta y seguiremos tu blog

Responder



<https://disenowebakus.net/imagenes/logo-akus.jpg> · mayo 2

muchas gracias esta buenísimo su blog me podría ayudar en el pronóstico de la oferta y la demanda de maíz basado en redes neuronales

Responder



Na8 · mayo 3

Hola, si, cuéntame más del proyecto e intentaré ayudar, aunque debo decir que casi casi no tengo tiempo disponible actualmente

Responder

Deja un comentario

Introduce aquí tu comentario...

Visita nuestra Guía de Aprendizaje

Buscar

Search ...

Contacto

Suscripción

Recibe los artículos de Aprende Machine Learning en tu casilla de correo. Cada 15 días y sin Spam!

Email:

ENVIAR

Obtén un **navegador compatible** para conseguir un reto reCAPTCHA.

¿Por qué tengo que hacer esto?