

Armando Rios Lastiri

Responde a las siguientes preguntas de forma individual. Todos los incisos tienen el mismo peso.

1. ¿A qué se denomina Aprendizaje Profundo y en qué se diferencia del Aprendizaje de Automatizado?

El aprendizaje profundo es un subconjunto del aprendizaje automatizado el cual a su vez es subconjunto de la inteligencia artificial. En el aprendizaje profundo se diseñan algoritmos en capas para crear una "red neuronal artificial" que puede aprender y tomar decisiones inteligentes por sí misma. En el aprendizaje profundo, en lugar de enseñarle a una computadora una lista enorme de reglas para resolver un problema, le damos un modelo que pueda evaluar ejemplos y una pequeña colección de instrucciones para modificar el modelo cuando se produzcan errores. En el aprendizaje automático se requiere de mayor intervención humana para ajustar los modelos. También existen otras diferencias como se muestran a continuación:

Aprendizaje Automático	Aprendizaje Profundo
Puede entrenar con menores datos de entrenamiento	Requiere grandes conjuntos de datos para el entrenamiento
Toma menos tiempo entrenar	Toma más tiempo entrenar
Se entrena en CPU	Se entrena en GPU para un entrenamiento adecuado
La salida es de forma numérica para aplicaciones de clasificación y puntuación.	La salida puede ser de cualquier forma, incluidos elementos de forma libre como texto libre y sonido.
Capacidad de ajuste limitada para el ajuste de hiperparámetros	Se puede ajustar de varias formas.

2. ¿Qué es la ingeniería de características y cómo se relaciona con el Aprendizaje Profundo?

La ingeniería de características trata de transformar los datos de entrenamiento y aumentarlos con funciones adicionales, para hacer que los algoritmos de aprendizaje automático sean más efectivos. Con el aprendizaje profundo, uno puede comenzar con datos sin procesar, ya que la red neuronal creará características automáticamente cuando aprenda. Pero esto no significa que el preprocesamiento de datos, la extracción de características y la ingeniería de características sean totalmente irrelevantes cuando se usa el aprendizaje profundo.

3. ¿Qué es una capa completamente conectada?

En una capa completamente conectada, cada una de las salidas de la capa anterior está conectada con un peso a cada neurona de la capa. El número de pesos en este tipo de capa aumenta rápidamente cuando aumenta el número de entradas o el número de neuronas en la capa. El entrenamiento se vuelve más difícil cuando aumenta la cantidad de pesos por lo tanto, la capa completamente conectada solo se usa cuando el número de entradas es pequeño.

4. ¿En qué consiste el algoritmo del descenso por gradiente?

El método del descenso del gradiente es un algoritmo de optimización que permite converger hacia el valor mínimo de una función mediante un proceso iterativo. En aprendizaje automático básicamente se utiliza para minimizar una función que mide el error de predicción del modelo en el conjunto de datos. A esta función de error se le suele denominar función de costo e identificar con $J(\theta)$, en donde θ hace referencia a los parámetros del modelo.

Para identificar el mínimo de la función el método del descenso del gradiente calcula la derivada parcial respecto a cada parámetro en el punto de evaluación. La derivada indica el valor y sentido en que se encuentra el mínimo más próximo. Este puede ser tanto un mínimo local como global, el método no los puede diferenciar. El resultado de la derivada se le resta a cada uno de los parámetros multiplicado por la velocidad de aprendizaje α . La velocidad de aprendizaje generalmente tiene un valor entre 0 y 1 e indica lo rápido que converge el algoritmo. Es importante notar que es necesario seleccionar un valor adecuado. Un valor demasiado bajo puede provocar que nunca se alcance el mínimo. Por otro lado, un valor lo demasiado alto podría saltarse el mínimo.

5. ¿Cuál es la relación entre el algoritmo del descenso por gradiente y retropropagación?

La retropropagación es el algoritmo que se utiliza para calcular el gradiente de la función de pérdida con respecto a los parámetros de la red neuronal. El descenso de gradiente es el algoritmo de optimización que se utiliza para encontrar parámetros que minimicen la función de pérdida.

6. ¿Qué es el desvanecimiento de gradiente y cuándo se presenta?

El problema del desvanecimiento del gradiente ocurre cuando el algoritmo de retropropagación retrocede a través de todas las neuronas de la red neuronal para actualizar sus pesos. La función de costo calculada en una capa profunda de la red neuronal se utilizará para cambiar los pesos de las neuronas en las capas menos profundas. El gradiente calculado en lo profundo de la red se desvanece a medida que se retrocede a través de la red y los pesos ya no cambian.

7. ¿Cuáles son las ventajas de la función ReLU respecto a la Sigmoide?

En la función ReLU no se tiene el problema del desvanecimiento del gradiente además es computacionalmente más eficiente que la sigmoide ya que no tiene que realizar operaciones exponenciales que son muy costosas. Otra ventaja que tiene el uso de la función ReLU es que las redes tienden a mostrar mejor convergencia que la función sigmoide.

8. Dada una imagen de entrada de 28x28 píxeles, ¿cuál es la dimensión de salida de una capa convolucional 2D si se usa un filtro de tamaño 3x3, un salto de 2x2 y no se usa relleno? ¿y si se usa relleno?

Sin relleno:

$$H_0 = (28-3)/2+1 = \text{floor}(13.5) = 13$$

$$W_0 = (28-3)/2+1 = \text{floor}(13.5) = 13$$

$$C_0 = 1$$

Con relleno de 1:

$$H_0 = (28-3+2*1)/2+1 = \text{floor}(14.5) = 14$$

$$W_0 = (28-3+2*1)/2+1 = \text{floor}(14.5) = 14$$

$$C_0 = 1$$

9. ¿Cómo es posible simular el campo receptivo de un filtro de convolución de 5x5 con dos filtros de 3x3? ¿Qué pasa con el número de parámetros?

10. ¿Por qué se dice que la convolución es invariante a traslación?

La convolución proporciona equivalencia a la traslación lo cual significa que si mueve la imagen hacia la derecha, también lo hace su capa de características producida por la convolución. El pooling proporciona invarianza a la traslación, pero solo aproximadamente. Una capa de pooling como la max pooling devuelve el valor más grande en su campo receptivo. Si el valor más grande se mueve hacia la derecha y todavía está dentro del campo receptivo, la capa de pooling aún genera ese valor más grande por lo que se volvió invariable al moverlo hacia la derecha.