DECONVOLUCIÓN CIEGA DE IMÁGENES CON TÉCNICAS DE APRENDIZAJE PROFUNDO

1.- Idea general de aprendizaje automático

El Aprendizaje Profundo (Deep Learning) constituye un campo interdisciplinario que aborda principios matemáticos, estadísticos y de ciencias de la computación, con el propósito de construir de modelos con la capacidad de adquirir representaciones complejas a partir de conjuntos de datos; Este enfoque se orienta hacia la emulación del funcionamiento cerebral humano, capacitando a los sistema computacionales para asimilar y procesar información proveniente de diversas fuentes, lo que les permite detectar patrones , realizar predicciones y mejorar su desempeño a través de la experiencia adquirida mediante los inputs recibidos.

1.1 Aprender de los datos

El caso de uso más sencillo para un modelo entrenado a partir de datos es cuando se tiene acceso a una señal x, por ejemplo, consideremos el escenario donde se tiene una imagen de una matrícula de un vehículo (x), a partir de la cual se quiere predecir una cantidad y, como la cadena de caracteres escritos en la matrícula.

En situaciones donde los datos (x) son de alta dimensionalidad y provienen de entornos no controlados, encontrar una relación analítica precisa entre la entrada (x) y la salida (y) puede ser extremadamente complejo.

Lo que sí se puede hacer es recopilar un gran conjunto de entrenamiento ($\mathcal{D}$) que contiene pares de datos de entrada y salida (x_n,y_n), e idear un modelo paramétrico f, con parámetros entrenables (w) que modulan su comportamiento El objetivo del entrenamiento es encontrar valores de parámetros (w“) que minimice una función de pérdida ($\mathcal{L}$(w)), que cuantifica la discrepancia entre las predicciones del modelo y las salidas reales en el conjunto del entrenamiento.

1.2 Underfitting y Overfitting

Una consideración clave es encontrar un equilibrio entre la capacidad del modelo (su flexibilidad y habilidad para ajustarse a datos diversos) y la cantidad y calidad de los datos de entrenamiento.

Underfitting: Cuando la capacidad es insuficiente, el modelo no puede ajustarse a los datos (es decir, el modelo es demasiado simple para capturar la estructura subyacente de los datos).

Overfitting: Cuando la cantidad de datos es insuficiente, el modelo suele aprender características específicas de los ejemplos de entrenamiento, lo que se traduce en un excelente rendimiento durante el entrenamiento, a costa de un peor ajuste a la estructura global de los datos y un rendimiento deficiente a las nuevas entradas (es decir, el modelo

se ajusta demasiado a los datos de entrenamiento, perdiendo capacidad de generalización).

El arte del aprendizaje automático radica en diseñar modelos que no sean demasiado flexibles pero capaces de ajustarse a los datos sin overfitting.

1.3 Categorías de modelos

Podemos categorizar los modelos de aprendizaje en tres grandes grupos_

- Regresión: Tiene como objetivo predecir valores continuos basados en variables independientes. Por ejemplo, posición geométrica de un objeto dada una señal de entrada x; U otro ejemplo: predecir el precio de una casa basándose en características como el área, el número de habitaciones, la ubicación, etc.
- Clasificación: El objetivo es asignar una etiqueta/categoría a una instancia de datos basándose en sus características. Por ejemplo, para identificar el tipo de animal en una imagen.
- Modelado de densidad: Tiene como objetivo modelar la función de densidad de probabilidad de los datos μX para describir la distribución de los datos (es decir, el objetivo es entender la estructura subyacente de los datos y modelar cómo se distribuyen en el espacio de características). Esto puede ser útil en la generación de imágenes, donde se desea crear nuevas imágenes que se parezcan a las de un conjunto de datos de entrenamiento.

Tanto la regresión como la clasificación se refieren generalmente al aprendizaje supervisado, ya que el valor a predecir, que se requiere como objetivo durante el entrenamiento, debe ser proporcionado, mientras la modelización de densidad se considera aprendizaje no supervisado, ya que basta con tomar datos existentes sin necesidad de producir una verdad de referencia asociada.

Estas tres categorías no son mutuamente excluyentes, existen multitud de problemas los cuales se abordan utilizando más de una de estas categorías.

2.- Regresión Lineal

En el campo del aprendizaje automático la regresión lineal emerge como una de las herramientas fundamentales para modelar y comprender relaciones entre variables. En este contexto, el objetivo principal es encontrar una línea recta que se ajuste de manera óptima a un conjunto de datos dispersos en un plano, lo que implica la predicción de un valor numérico a partir de variables independientes (aquí es donde radican los problemas de la regresión).

Consideremos un ejemplo práctico: supongamos que se desea predecir el precio de viviendas en función de su superficie y su antigüedad. Para desarrollar un modelo de predicción de precios de la vivienda, necesitamos disponer de datos que incluyan el precio de venta, la superficie y la antigüedad de las casas. En la terminología del aprendizaje automático, el conjunto de datos se denomina conjunto de datos de entrenamiento o conjunto de entrenamiento, y cada fila (que contiene los datos

correspondientes a una venta) se denomina ejemplo (o punto de datos, instancia, muestra). La variable que intentamos predecir (el precio) se denomina etiqueta (u objetivo). Las variables (superficie y antigüedad) en las que se basan las predicciones se denominan características (o covariables).

2.1 Conceptos Básicos

La regresión lineal es a la vez la más sencilla y la más popular de las herramientas estándar para abordar los problemas de regresión. Este método, que tiene sus raíces en el siglo XIX (en los trabajos pioneros de Gauss y Legendre), se basa en varios supuestos fundamentales que deben cumplirse para su aplicación efectiva.

En primer lugar, se supone que la relación entre las características (x) y el objetivo (y) es aproximadamente lineal, lo que implica que la media condicional de (y) dado (x) E[Y|X=x] puede expresarse como una suma ponderada de las características (x). Esto significa que el valor esperado de la variable dependiente varia linealmente con respecto a las variables independientes. Esta configuración presupone el "ruido de observación"; esto se refiere a que el valor objetivo pueda desviarse de su valor esperado. Este ruido puede deberse a diversas razones, como errores de medición, fluctuaciones aleatorias en el sistema que estamos estudiando, etc. Para abordar este ruido se impone el supuesto de que dicho ruido se comporta bien, siguiendo una distribución gaussiana o normal.

En lo que se refiere a la notición, 'n' se utiliza para indicar el número total de ejemplos en el conjunto de datos. Los superíndices se utilizan para enumerar las muestras y los objetivos ($x^{(i)}$), y subíndices para indexar las coordenadas ($x_{j}^{(i)}$). Por ejemplo, si estamos analizando el desempeño académico de estudiantes y cada fila de nuestro conjunto de datos representa a un estudiante, entonces $x^{(i)}$ sería una fila particular que corresponde a un estudiante específico. Ahora, $x_{j}^{(i)}$ indica el valor de una característica específica para esa observación en particular. Aquí, 'j' representa el índice de la característica. Por ejemplo, si estamos analizando el desempeño académico de los estudiantes y tenemos características como horas de estudio, puntaje en matemáticas, puntaje en ciencias, etc., entonces $x_{j}^{(i)}$ sería el valor de la característica 'j' (por ejemplo, horas de estudio, puntaje en matemáticas, etc.) para la observación 'i' (por ejemplo, el estudiante 'i').

2.2 Modelo matemático

El supuesto de linealidad significa que se asume que la relación entre las características (variable independiente) y el objetivo (variable dependiente) es lineal. Esto es, que el valor esperado del objetivo puede expresarse como una suma ponderada de las características. El modelo en su forma compacta es el siguiente:

$$\hat{y} = \mathbf{w}^T \cdot \mathbf{x} + b$$

En la ecuación x es el vector de características, w se denomina vector de pesos/ponderaciones, y b se denomina sesgo. La ponderación determina la influencia/peso de cada característica en la predicción, mientras que el sesgo determina el valor de estimación cuando todas las características son 0.

Para tratar las características de todo nuestro conjunto de datos de n ejemplos de manera eficiente se utiliza la matriz de diseño, donde X contiene una fila para cada ejemplo y una columna para cada característica, haciendo que las predicciones ^y se expresen mediante un producto matriz-vector:

$$y\text{^}=\mathbf{X}\cdot\mathbf{w}+b$$

Dadas las características de un conjunto de datos de entrenamiento 'X' y las correspondientes etiquetas (conocidas) 'y', el objetivo de la regresión lineal es encontrar el vector de pesos 'w' y el término de sesgo 'b' de forma que, dadas las características de un nuevo ejemplo de datos muestreado a partir de la misma distribución que 'X' la etiqueta del nuevo ejemplo se prediga (en principio) con el menor error, es decir, que, en promedio, los valores del vector de pesos 'w' y el sesgo 'b' hagan que las predicciones de nuestro modelo se ajusten lo más posible a los valores reales observados en los datos.

Antes de buscar los mejores parámetros w y b necesitaremos:

(i) Una medida de calidad.
(ii) Un procedimiento para actualizar el modelo.

2.3 Función de pérdida

Para ajustar nuestro modelo a los datos, se necesita definir una medida de adecuación/ajuste; esta es la función de pérdida, la cual cuantifica la diferencia entre los valores reales y las predicciones del objetivo. En el contexto de la regresión lineal, la función de pérdida más común es el error cuadrático medio (MSE):

$$l^{(i)}(\mathbf{w},b)=\frac{1}{2}\left(\hat{y}^{(i)}-y^{(i)}\right)^2$$

- $y\text{^}(i)$ es la predicción del modelo para el ejemplo i.
- $y(i)$ es el valor real correspondiente al ejemplo i.

La pérdida suele ser un número no negativo en el que los valores más pequeños son mejores y las predicciones perfectas incurren en una pérdida de 0.

Es importante destacar que la naturaleza cuadrática del MSE puede hacer que el modelo evite grandes errores, pero sea excesivamente sensible a los datos anómalos (es decir, el minimizar esta función puede llevar a un ajuste excesivo a puntos atípicos en el conjunto de datos). Para medir la calidad de un modelo en todo el conjunto de datos de n ejemplos, basta con promediar las pérdidas en el conjunto de entrenamiento:

$$L(\mathbf{w},b)=\frac{1}{n}\sum_{i=1}^{n}l^{(i)}(\mathbf{w},b)=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^\top\mathbf{x}^{(i)}+b-y^{(i)}\right)^2.$$

2.4 Solución analítica

La regresión lineal ofrece una manera sencilla y elegante de calcular los valores óptimos de los parámetros del modelo que minimicen la función de pérdida a través de la siguiente. Primero:

$$L(w)=||y-Xw||^2$$

podemos subsumir el sesgo b en el parámetro w añadiendo una columna a la matriz de diseño formada por toda por 1s. Entonces nuestro problema de predicción es minimizar ||y - Xw||^2. Mientras la matriz de diseño X tenga rango completo (ninguna característica depende linealmente de las demás), sólo habrá un punto crítico en la superficie de pérdida, que corresponde al mínimo de la pérdida en todo el dominio. Si se toma la derivada de la pérdida con respecto a w e igualándola a cero se obtiene:

$$\partial_{\mathbf{w}}\|\mathbf{y}-\mathbf{X}\mathbf{w}\|^2 = 2\mathbf{X}^\top(\mathbf{X}\mathbf{w}-\mathbf{y}) = 0 \text{ and hence } \mathbf{X}^\top\mathbf{y} = \mathbf{X}^\top\mathbf{X}\mathbf{w}.$$

Resolviendo para w, nos proporciona la solución óptima para el problema de optimización:

$$\mathbf{w}^* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

Nótese que esta solución solo será única cuando la matriz X^тX es invertible.

2.5 Descenso de Gradiente Estocástico por Mini-Lotes

Esta técnica es crucial para optimizar modelos que no pueden ser resueltos de manera analítica y pueden resultar difíciles de optimizar. El descenso de gradiente es el algoritmo principal para reducir iterativamente el error mediante la actualización de los parámetros en la dirección que disminuye incrementalmente la función de pérdida.

La aplicación más básica del descenso de gradiente es calcular la derivada de la función de pérdida, que es un promedio de las pérdidas calculadas en cada ejemplo del conjunto de datos. Sin embargo, en la práctica realizar una actualización completa en cada paso puede ser extremadamente lento, especialmente si hay mucha redundancia en los datos de entrenamiento.

El descenso de gradiente estocástico (SGD) es una solución eficaz (incluso para grandes conjuntos de datos) que consiste en considerar un único ejemplo a la vez para actualizar los parámetros. Aunque eficiente, el SGD tiene desventajas tanto computacionales como estadísticas, como el tiempo necesario para procesar una muestra a la vez y la dificultad para aplicar ciertas técnicas como la normalización por lotes.

Para abordar estos problemas se propone una estrategia intermedia, el descenso de gradiente estocástico mini-batch, donde en lugar de tomar un lote completo o una sola muestra a la vez, tomamos un mini-lote de observaciones.

El funcionamiento de este modelo es el siguiente:

(i) Inicializa los valores de los parámetros del modelo de forma aleatoria.
(ii) Muestrea iterativamente minilotes aleatorios de los datos, actualizando los parámetros en la dirección del gradiente negativo de la pérdida promedio en el minibatch (el gradiente indica la dirección y la magnitud en la que la pérdida cambia más rápidamente con respecto a los parámetros del modelo).

Después del entrenamiento, los parámetros del modelo se registran para su evaluación en un conjunto de datos de validación separado.

Aunque el algoritmo converge lentamente hacia los minimizadores de pérdida, normalmente no los encontrará exactamente en un número finito de pasos, sin embargo, en la práctica, el objetivo es encontrar cualquier conjunto de parámetros que conduzca a predicciones precisas sobre datos nunca antes vistos.

2.6 Vectorización para velocidad

Durante el proceso de entrenamiento de modelos de regresión lineal, la eficiencia computacional es crucial para manejar grandes conjuntos de datos de manera efectiva. Cuando entrenamos nuestros modelos, normalmente queremos procesar mini-lotes enteros de ejemplos simultáneamente. Para hacerlo de forma eficiente, es necesario vectorizar los cálculos.

La vectorización es una técnica fundamental en la programación computacional que consiste en realizar operaciones sobre vectores o matrices enteras en lugar de elementos individuales. En el contexto de Python y el aprendizaje automático, la vectorización se logra utilizando bibliotecas especializadas de álgebra lineal, como 'PyTorch', que están optimizadas para ejecutar operaciones en matrices de manera eficiente.

Para ilustrar la importancia de la vectorización, consideremos dos métodos para sumar vectores de alta dimensionalidad:

1er método) Iteramos sobre cada elemento de los vectores utilizando un bucle for en Python y sumamos los elementos correspondientes uno a uno.

```
n = 10000
a = torch.ones(n)
b = torch.ones(n)
c = torch.zeros(n)
t = time.time()

for i in range(n):
c[i] = a[i] + b[i]

f'{time.time() - t:.5f} sec'
```

```
'0.17802 sec'
```

2do método) aprovechamos las capacidades de vectorización de la biblioteca PyTorch para sumar los vectores en una sola operación '+'.

```
t = time.time()
d = a + b
f'{time.time() - t:.5f} sec'
```

```
'0.00036 sec'
```

Al comparar los tiempos de ejecución de ambos métodos, observamos una diferencia significativa en la eficiencia a favor del método donde utilizamos operaciones vectorizadas. Este aumento en la velocidad de ejecución puede ser del orden de magnitud, lo que demuestra la importancia de la vectorización para optimizar el rendimiento de los algoritmos de regresión lineal. Además, también reduce la complejidad del código y aumenta su portabilidad al trasladar la carga computacional a las bibliotecas de álgebra lineal.

2.7 La distribución normal y la pérdida al cuadrado

La distribución normal o gaussiana es una de las distribuciones de probabilidad más importantes en la teoría estadística. Esta viene determinada por dos parámetros: la media μ y la varianza σ2 (desviación típica σ); y viene dada por la fórmula:

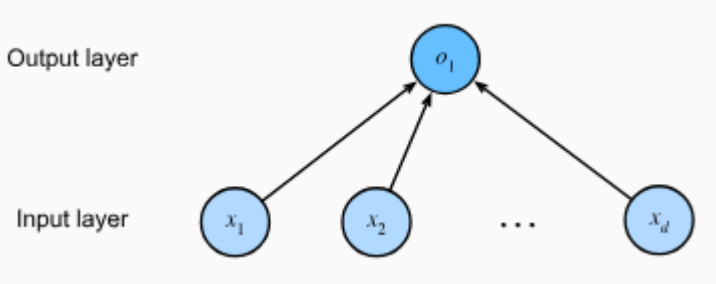
$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right).$$

La regresión lineal busca modelar la relación entre una variable dependiente Y y una o más variables independientes X mediante una función lineal. En el contexto de la regresión lineal, se asume que las observaciones están sujetas a errores aleatorios (es decir, las diferencias entre los valores observados y los valores predichos por el modelo) que siguen una distribución normal. Es decir, se supone que los errores de predicción ε tienen una distribución normal con una media de cero (se espera que los errores de predicción se distribuyan alrededor de cero) y una varianza constante σ^2 (la dispersión de los errores es constante en todo el rango de los valores de las variables independientes).

La elección de la pérdida al cuadrado en la regresión lineal se justifica teóricamente mediante el principio de máxima verosimilitud (es decir, buscar aquellos valores de los parámetros del modelo que hacen que nuestros datos sean lo más probables posible). Según este principio, los mejores estimadores de los parámetros del modelo son aquellos que maximizan la probabilidad de observar los datos dados los parámetros del modelo. En el caso de la regresión lineal, maximizar la verosimilitud es equivalente a minimizar la suma de los cuadrados de los errores residuales, ya que se asume que los errores siguen una distribución normal. Por lo tanto, la minimización de la pérdida al cuadrado en la regresión lineal conduce a estimaciones de parámetros que son consistentes con el enfoque de máxima verosimilitud.

2.8 Regresión Lineal como red neuronal

La siguiente figura representa la regresión lineal como una red neuronal. El diagrama destaca el patrón de conectividad, por ejemplo, cómo se conecta cada entrada $x_1,\ldots,x_d$ (es decir las diferentes características o variables independientes) a la salida 'o_1' (la predicción), pero no los valores específicos que toman los pesos o los sesgos.



En resumen, en este ilustración podemos pensar en la regresión lineal como una red neuronal totalmente conectada de una sola capa.