

# Evaluación del desempeño de los modelos de aprendizaje profundo

# Introducción

Hay muchas decisiones que tomar al diseñar y configurar sus modelos de aprendizaje profundo. La mayoría de estas decisiones deben resolverse empíricamente mediante prueba y error y evaluándolas con datos reales. Como tal, es de vital importancia tener una forma sólida de evaluar el rendimiento de su red neuronal y modelos de aprendizaje profundo.

# Introducción

¿Qué se puede utilizar para evaluar el rendimiento del modelo con Keras?

¿Cómo evaluar un modelo de Keras utilizando un conjunto de datos de verificación automática?

¿Cómo evaluar un modelo de Keras usando un conjunto de datos de verificación manual?

¿Cómo evaluar un modelo de Keras mediante la validación cruzada de k-fold?

# Evaluar empíricamente las configuraciones de red

- Hay una gran cantidad de decisiones que debe tomar al diseñar y configurar sus modelos de aprendizaje profundo. Muchas de estas decisiones se pueden resolver copiando la estructura de las redes de otras personas y utilizando heurísticas.
- En última instancia, la mejor técnica es diseñar pequeños experimentos y evaluar empíricamente las opciones utilizando datos reales.
- Esto incluye decisiones de alto nivel como el número, tamaño y tipo de capas en su red. También incluye las decisiones de nivel inferior como la elección de la función de pérdida, las funciones de activación, el procedimiento de optimización y el número de épocas.

# Evaluar empíricamente las configuraciones de red

- El aprendizaje profundo se usa a menudo en problemas que tienen conjuntos de datos muy grandes. Eso es decenas de miles o cientos de miles de instancias. Como tal, necesita tener un arnés de prueba robusto que le permita estimar el desempeño de una configuración dada en datos invisibles y comparar confiablemente el desempeño con otras configuraciones.

# División de datos

La gran cantidad de datos y la complejidad de los modelos requieren tiempos de entrenamiento muy largos. Como tal, es típico utilizar una simple separación de datos en conjuntos de datos de entrenamiento y prueba o conjuntos de datos de entrenamiento y validación. Keras proporciona dos formas convenientes de evaluar sus algoritmos de aprendizaje profundo de esta manera:

1. Utilizar un conjunto de datos de verificación automática.
2. Utilizar un conjunto de datos de verificación manual.

# Conjunto de datos de verificación automática

- Keras puede separar una parte de sus datos de entrenamiento en un conjunto de datos de validación y evaluar el desempeño de su modelo en ese conjunto de datos de validación en cada época. Puede hacer esto configurando el argumento de división de validación en la función `fit()` en un porcentaje del tamaño de su conjunto de datos de entrenamiento. Por ejemplo, un valor razonable podría ser 0,2 o 0,33 para el 20% o el 33% de los datos de entrenamiento separados para su validación.

# Conjunto de datos de verificación automática

- Keras puede separar una parte de sus datos de entrenamiento en un conjunto de datos de validación y evaluar el desempeño de su modelo en ese conjunto de datos de validación en cada época. Puede hacer esto configurando el argumento de división de validación en la función `fit()` en un porcentaje del tamaño de su conjunto de datos de entrenamiento. Por ejemplo, un valor razonable podría ser 0,2 o 0,33 para el 20% o el 33% de los datos de entrenamiento separados para su validación.



# Conjunto de datos de verificación manual

- Keras también le permite especificar manualmente el conjunto de datos que se utilizará para la validación durante el entrenamiento. Podemos usar la función `train test split ()` de la biblioteca de aprendizaje automático Python `scikit-learn` para separar nuestros datos en un conjunto de datos de entrenamiento y prueba. Usamos el 67% para entrenamiento y el 33% restante de los datos para validación. El conjunto de datos de validación se puede especificar a la función `fit ()` en Keras mediante el argumento de datos de validación. Toma una tupla de los conjuntos de datos de entrada y salida.

# Validación cruzada manual k-Fold

- Las clases `KerasClassifier` y `KerasRegressor` en Keras toman un argumento `build_fn` que es el nombre de la función a llamar para crear su modelo. Debes definir una función llamada como quieras que defina tu modelo, lo compile y lo devuelva.
- Se pasa este nombre de función a la clase `KerasClassifier` mediante el argumento `fn_build`. También pasamos argumentos adicionales de `epochs` 150 y `batch_size`. Estos se agrupan automáticamente y se pasan a la función `fit()` que es llamada internamente por la clase `KerasClassifier`. Por ejemplo usamos `scikit-learn StratifiedKFold` para realizar una validación cruzada estratificada de 10 veces. Esta es una técnica de remuestreo que puede proporcionar una estimación sólida del rendimiento de un modelo de aprendizaje automático en datos no vistos. Usamos la función `cross_val_score()` de `scikit-learn` para evaluar nuestro modelo usando el esquema de validación cruzada e imprimir los resultados.