

Tareas de Apoyo en Proyectos de Deep Learning

Juan Gamarra Moreno

Guardar el modelo con serialización

Dado que los modelos de aprendizaje profundo pueden tardar horas, días e incluso semanas en entrenarse, es importante saber cómo puede guardar sus modelos de Keras en un archivo y volver a cargarlos para hacer predicciones. Resulta apropiado, saber:

- Cómo guardar y cargar pesos de modelos de Keras en archivos con formato HDF5.
- Cómo guardar y cargar la estructura del modelo Keras en archivos JSON.
- Cómo guardar y cargar la estructura del modelo de Keras en archivos YAML.

Guardar el modelo con serialización

- Keras separa las preocupaciones de guardar la arquitectura de su modelo y guardar los pesos de su modelo.
- Los pesos de los modelos se guardan en formato HDF5. Este es un formato de cuadrícula ideal para almacenar matrices de números multidimensionales.
- La estructura del modelo se puede describir y guardar (y cargar) utilizando dos formatos diferentes: JSON y YAML.

Guardar el modelo con serialización

Formato HDF5

El formato de datos jerárquico o HDF5 para abreviar es un formato de almacenamiento de datos flexible y es conveniente para almacenar grandes matrices de valores reales, como tenemos en los pesos de las redes neuronales.

Es posible que deba instalar el soporte de Python para el formato de archivo HDF5. Puede hacer esto usando su sistema de administración de paquetes de Python preferido e instalar h5py

Guardar el modelo con serialización

Guardar como Jason

El formato de datos jerárquico o HDF5 para abreviar es un formato de almacenamiento de datos flexible y es conveniente para almacenar grandes matrices de valores reales, como tenemos en los pesos de las redes neuronales.

Es posible que deba instalar el soporte de Python para el formato de archivo HDF5. Puede hacer esto usando su sistema de administración de paquetes de Python preferido e instalar h5py

Guardar el modelo con serialización

Guardar como Jason

JSON es un formato de archivo simple para describir datos jerárquicamente. Keras proporciona la capacidad de describir cualquier modelo usando formato JSON con una función `to_json()`. Esto se puede guardar en un archivo y luego cargar a través de la función `model_from_json()` que creará un nuevo modelo a partir de la especificación JSON.

Guardar el modelo con serialización

Guardar como Jason (cont.)

Los pesos se guardan directamente del modelo usando la función `save_weights()` y luego se cargan usando la función simétrica `load_weights()`.

Guardar el modelo con serialización

Guardar como Jason (cont.)

Por ejemplo se puede entrenar y evaluar un modelo usando un conjunto de datos. Luego, la estructura del modelo se convierte a formato JSON y se guarda. Los pesos de la red se escriben en un archivo separado.

Después los datos del modelo y peso se cargan desde los archivos guardados y se crea un nuevo modelo. Es importante compilar el modelo cargado antes de usarlo. Esto es para que las predicciones hechas usando el modelo puedan usar el cálculo eficiente apropiado del backend de Keras.

Guardar el modelo con serialización

Guardar como YAML

El procedimiento es muy similar a guardar como JSON, excepto que se usa el formato YAML para la especificación del modelo. El modelo se describe usando YAML, se guarda en un archivo con extensión yaml y luego se carga en un nuevo modelo a través de la función `model_from_yaml()`. Los pesos se manejan en un formato HDF5 de la misma forma que en el caso de JSON.

Guardar el modelo con serialización

Guardar juntos Pesos y Arquitectura

Keras también admite una interfaz más sencilla para guardar los pesos del modelo y la arquitectura del modelo juntos en un solo archivo H5. Guardar el modelo de esta manera incluye todo lo que necesitamos saber sobre el modelo:

- Pesos del modelo.
- Arquitectura modelo.
- Detalles de la compilación del modelo (pérdidas y métricas).
- Estado del optimizador de modelo.

Guardar el modelo con serialización

Guardar juntos Pesos y Arquitectura (cont.)

Esto significa que podemos cargar y usar el modelo directamente, sin tener que volver a compilarlo. Esta es la forma preferida de guardar y cargar modelos con Keras.

Así, se puede guardar su modelo llamando a la función `save()` y especificando el nombre del archivo. Por ejemplo se puede entrenar un modelo, evaluarlo y guardarlo en un archivo con extensión `h5`.

Guardar el modelo con serialización

Guardar juntos Pesos y Arquitectura (cont.)

Su modelo guardado se puede cargar más tarde llamando a la función `load_model()` y pasando el nombre del archivo. La función devuelve el modelo con la misma arquitectura y pesos.

Puntos de Control

Los modelos de aprendizaje profundo pueden tardar horas, días o incluso semanas en entrenarse y si la ejecución del entrenamiento se detiene inesperadamente, puede perder mucho trabajo. Así es importante:

- Controlar los modelos de redes neuronales durante el entrenamiento.
- Establecer un punto de control en cada mejora de un modelo durante el entrenamiento.
- Establecer un punto de control en el mejor modelo observado durante el entrenamiento.

Puntos de Control

Estableciendo puntos de control

El punto de control de aplicaciones es una técnica de tolerancia a fallos para procesos de larga ejecución. Es un enfoque en el que se toma una instantánea del estado del sistema en caso de falla del sistema. Si hay un problema, no todo está perdido. El punto de control puede usarse directamente o usarse como punto de partida para una nueva ejecución, retomando donde lo dejó. Al entrenar modelos de aprendizaje profundo, el punto de control captura los pesos del modelo. Estos pesos se pueden utilizar para hacer predicciones tal cual o como base para continuar el entrenamiento.

Puntos de Control

Estableciendo puntos de control (cont.)

La biblioteca Keras permite establecer puntos de control mediante una API de devolución de llamada. La clase `ModelCheckpoint` le permite definir dónde controlar los pesos del modelo, cómo se debe nombrar el archivo y bajo qué circunstancias hacer un punto de control del modelo. La API le permite especificar qué métrica monitorear, como la pérdida o precisión en el conjunto de datos de entrenamiento o validación. Puede especificar si buscar una mejora para maximizar o minimizar la puntuación.

Puntos de Control

Estableciendo puntos de control (cont.)

Finalmente, el archivo que usa para almacenar los pesos puede incluir variables como el número de época o la métrica.

La instancia de ModelCheckpoint se puede pasar al proceso de entrenamiento cuando se llama a la función `fit()` en el modelo.

Tenga en cuenta que necesita instalar la biblioteca `h5py`

Puntos de Control

PC: Mejoras en el modelo de red neuronal

Un buen uso de los puntos de control es generar los pesos del modelo cada vez que se observa una mejora durante el entrenamiento.

Por ejemplo los puntos de control podrían estar configurados para guardar los pesos de la red solo cuando hay una mejora en la precisión de la clasificación en el conjunto de datos de validación (monitor = 'val_accuracy' y mode = 'max'). Los pesos se almacenan en un archivo que incluye la puntuación en el nombre de archivo por ejemplo.

Puntos de Control

PC: Mejoras en el modelo de red neuronal (cont.)

Esta es una estrategia de puntos de control muy simple. Puede crear muchos archivos de puntos de control innecesarios si la precisión de la validación sube y baja durante las épocas de entrenamiento. Sin embargo, se asegurará de que tenga una instantánea del mejor modelo descubierto durante su ejecución.

Puntos de Control

PC: Solo el mejor modelo de red neuronal

Una estrategia de punto de control más simple es guardar los pesos del modelo en el mismo archivo, si y solo si mejora la precisión de la validación. Esto se puede hacer fácilmente, cambiando el nombre del archivo de salida para que sea fijo (sin incluir información de puntuación o época). En este caso, las ponderaciones del modelo se escriben en el archivo solo si la exactitud de clasificación del modelo en el conjunto de datos de validación mejora con respecto a la mejor vista hasta ahora.

Puntos de Control

Cargar el modelo de red neuronal guardado

Ahora que ha visto cómo establecer puntos de control a sus modelos de aprendizaje profundo durante el entrenamiento, debe saber cómo cargar y usar un modelo con punto de control. El punto de control solo incluye los pesos del modelo. Asume que conoce la estructura de la red, este también se puede serializar en un archivo en formato JSON o YAML.

Puntos de Control

Cargar el modelo de red neuronal guardado (cont.)

Por ejemplo, si se conoce la estructura del modelo y se cargan los mejores pesos del experimento almacenados en el archivo guardado, el modelo se puede usar para hacer predicciones en los conjuntos de datos.