Chapter 19: Summary

Diedrick Darrell Darmadi - 1103223031

Bab 19 membahas bagaimana melatih dan menerapkan model TensorFlow dalam skala besar, sebuah tantangan penting dalam dunia machine learning modern ketika data semakin besar, model semakin kompleks, dan kebutuhan produksi semakin menuntut efisiensi tinggi. Bab ini menekankan bahwa membuat model yang akurat saja tidak cukup; model harus dapat dilatih dengan cepat, dioperasikan secara stabil, dan didistribusikan ke lingkungan nyata dengan performa yang dapat diandalkan. Oleh karena itu, bab ini memperkenalkan berbagai metode, perangkat, dan strategi untuk meningkatkan kecepatan pelatihan serta mempermudah proses deployment di lingkungan produksi.

Pembahasan dimulai dari konsep **TensorFlow graphs** dan bagaimana TensorFlow mengoptimalkan eksekusi komputasi. Dengan menggunakan tf.function(), kode Python dikompilasi menjadi graph sehingga memungkinkan optimasi seperti parallel execution, memory optimization, dan operation fusion. Mode graph ini jauh lebih efisien daripada eager execution, terutama ketika model dilatih pada dataset besar atau digunakan dalam inference berkecepatan tinggi. Bab ini menunjukkan bahwa kombinasi antara graph execution dan strategi hardware (GPU/TPU) adalah pondasi utama dalam melatih model skala besar.

Bagian selanjutnya mengulas **distributed training**, yaitu teknik melatih model pada banyak GPU, TPU, atau bahkan banyak server sekaligus. TensorFlow menyediakan *Distribution Strategies* seperti MirroredStrategy (multi-GPU lokal), MultiWorkerMirroredStrategy (pelatihan lintas server), dan TPUStrategy (untuk perangkat TPU). Dalam *data parallelism*, batch data dibagi ke perangkat-perangkat tersebut, masing-masing menghitung gradien secara paralel, dan gradien digabung kembali untuk memperbarui parameter global. Dengan strategi ini, waktu pelatihan dapat berkurang drastis, bahkan untuk model sangat besar seperti CNN atau Transformer.

Selain komputasi model, bab ini menekankan pentingnya **optimasi input pipeline** menggunakan tf.data. Tanpa pipeline yang efisien, GPU dapat idle dan memperlambat pelatihan. Teknik seperti prefetch(), interleave(), caching, dan penggunaan format **TFRecord** memberikan throughput data yang tinggi. Bab ini menunjukkan bahwa bottleneck sering kali berasal dari input pipeline, bukan dari model itu sendiri, sehingga optimasi data menjadi aspek kritis dalam training at scale.

Setelah model selesai dilatih, fokus berpindah ke tahap **deployment**, yaitu membuat model dapat digunakan dalam aplikasi nyata. TensorFlow menggunakan format standar **SavedModel**, yang menyimpan arsitektur, parameter, dan signature model sehingga dapat dipanggil dari bahasa apa pun, termasuk Python, C++, JavaScript, dan Java. SavedModel menjadi inti dari berbagai platform deployment TensorFlow, termasuk TensorFlow Serving, TensorFlow Lite, dan TensorFlow.js.

TensorFlow Serving dibahas sebagai solusi server-side inference yang sangat efisien dan scalable. Platform ini memungkinkan penyajian model secara cepat melalui REST API atau gRPC, mendukung rotasi versi model tanpa downtime, serta mampu melakukan batching dinamis untuk meningkatkan throughput. Dengan TensorFlow Serving, perusahaan dapat menjalankan model machine learning dalam layanan produksi berskala besar dengan stabilitas tinggi. Selain itu, bab ini juga menjelaskan deployment untuk perangkat mobile dan IoT melalui TensorFlow Lite

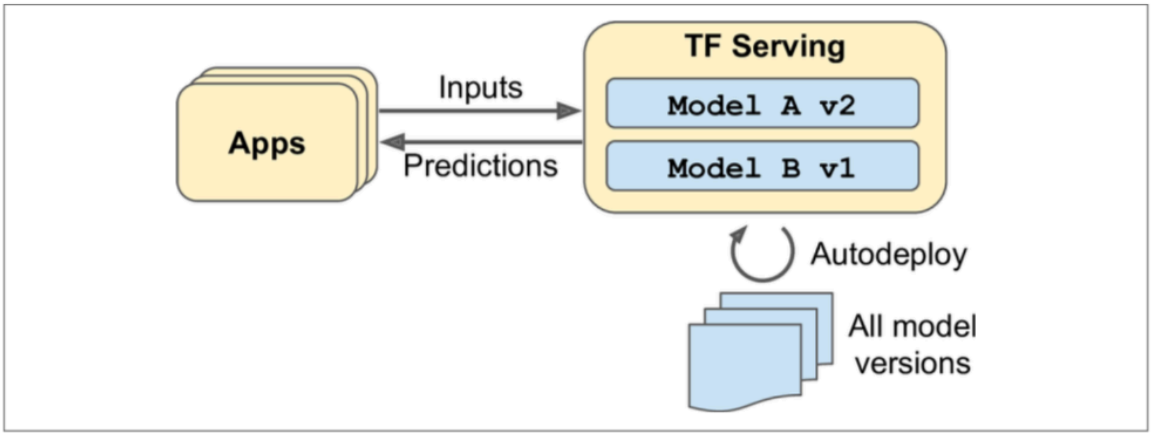

Figure 1 TF Serving can serve multiple models and automatically deploy the latest version of each model

Bagian akhir bab membahas praktik **MLOps** dan pengelolaan model di dunia nyata. Masalah seperti *data drift* dan *concept drift* dapat menyebabkan performa model menurun seiring waktu. Karena itu, monitoring model, logging, evaluasi berkala, dan retraining otomatis menjadi elemen penting. Bab ini juga menekankan pentingnya versioning untuk dataset dan model, serta integrasi pipeline otomatis agar sistem machine learning dapat diperbarui dengan lancar tanpa intervensi manual yang signifikan.

Secara keseluruhan, Bab 19 memberikan pemahaman menyeluruh mengenai bagaimana melatih model TensorFlow secara efisien dan bagaimana mendistribusikannya dengan aman dan optimal ke dalam sistem produksi. Dengan memahami teknik distribusi, optimasi pipeline, penggunaan hardware accelerators, dan strategi deployment profesional, pembaca dipersiapkan untuk menerapkan machine learning pada skala industri.