Model Compression

Distillation

Knowledge Distillation for AI

Shrinking the Models for Efficient Deployment

> 💡 In this presentation, we will explore the concept of knowledge distillation in AI models. We will discuss what it is, how it works, and the benefits it offers.

What is Knowledge Distillation?

- Knowledge distillation is a technique for transferring knowledge from a large, complex model (teacher) to a smaller, more efficient model (student).
- The teacher model is typically a pre-trained model that has been trained on a large dataset.
- The student model is a smaller model that is designed to be more efficient to run

> 💡 The teacher model has a lot of knowledge and experience (because it has been trained on a large dataset), and it can share this knowledge with the student model. The student model benefits from this knowledge by learning to perform well on a task, even though it is smaller and simpler than the teacher model.

Benefits of Knowledge Distillation

- Reduced model size: Knowledge distillation can be used to create smaller models that are more efficient to run.
- Improved performance: In some cases, knowledge distillation can actually improve the performance of a model.
- Knowledge transfer: This can be useful for tasks where labeled data is scarce.

We can use the knowledge transfer to improve a regional finance model that has not a big code base to extract information.

The Fundamentals

What do we need to know to do this tuning?

The Simple Explanation!

Loss Function in Knowledge Distillation

A loss function quantifies the model's prediction discrepancy. In knowledge distillation, two primary loss elements exist:

- **Hard target loss:** Measures the variance between the student model's predictions and the data's actual labels.
- **Soft target loss:** Compares the student model's predictions and the teacher model's soft targets.

Hard Target Loss: This is the traditional loss function, like mean squared error for regression or cross-entropy for classification. It compares the student's raw predictions (like a single class label) to the true labels in the data.

Soft Target Loss: This is a unique aspect of knowledge distillation. The teacher model's predictions are not just class labels, but probabilities for each class. The soft target loss encourages the student model to mimic these probabilities, capturing the teacher's nuanced understanding of the data beyond just the hard labels.

💡 A loss function is like a scorekeeper in the training game of machine learning models. It tells the model how well it's doing on a particular example by calculating the discrepancy between its predictions and the actual answers (targets).

Deep dive!

Loss Function in Knowledge Distillation

Hard Target Loss (Lce): This is commonly implemented as the categorical cross-entropy (CE) loss for classification tasks. It calculates the difference between the student model's raw class predictions (like predicting "cat") and the true labels (like "cat" or "dog") present in the data.

Soft Target Loss (DKL(q,p)): The KL divergence (DKL(q,p)) measures the difference between the student model's predicted probabilities (q) and the soft targets (probabilities) from the teacher model (p).

By combining these two loss terms (Lce and DKL(q,p)), we get the overall loss function (Lstudent) that guides the student model's training:

Lstudent = α * Lce + (1 - α) * DKL(q,p)

Here, α is a **hyperparameter** (weight) that controls the relative importance of the hard target loss and the soft target loss. Tuning α can influence the final performance of the student model.

Hyperparameters

Hyperparameters are essentially the settings that control the learning process of a machine learning model. They are the internal variables that the model learns from the data during training.

an analogy: Imagine training a chef. The chef (model) learns from the ingredients (data) and recipes (algorithms) to cook dishes. Hyperparameters are like the oven temperature, cooking time, or knife selection.

- Common examples of hyperparameters:
  - Learning rate
  - Number of hidden layers
  - Batch size

Loss Functions in Knowledge Distillation

Hard Target Loss (Lce)

This encourages the student to **agree with the true labels of the data**. In simpler terms, it penalizes the student for misclassifying examples.

Categorical cross-entropy (CE) is a common choice for Lce. It measures the difference between the student model's predictions (represented as probabilities for each class) and the true labels of the data.

Example: Imagine training a student model to classify between cat and dog images. The true label for a cat image would be [1, 0], where 1 represents "cat" and 0 represents "dog." The student model might predict probabilities like [0.8, 0.2], signifying an 80% chance of being a cat.

Soft Target Loss (DKL(q,p))

DKL(q,p) represents the Kullback-Leibler divergence between two probability distributions.

The soft targets hold more information than just the hard labels (class labels). They encode the teacher model's level of confidence for each class. Going back to our cat and dog classification example, imagine the teacher model predicting probabilities like [0.95, 0.05] for a particular image. This signifies a strong belief that the image is a cat (0.95 probability) and near certainty that it's not a dog (0.05 probability). The KL divergence loss nudges the student model to align its own probability distribution with the teacher's, even if they

both ultimately predict "cat" as the class. By following the teacher's lead on the confidence levels, the student can learn subtle patterns that might be crucial for accurate classification.

Finding Good Hyperparameters with Optuna

Hyperparameters

Hyperparameters are essentially the settings that control the learning process of a machine learning model. They are distinct from regular model parameters, which are the internal variables that the model learns from the data during training.

Other maybe not so good analogy: Imagine training a chef. The chef (model) learns from the ingredients (data) and recipes (algorithms) to cook dishes. Hyperparameters are like the oven temperature, cooking time, or knife selection. They influence how the chef (model) uses the ingredients and recipes to achieve the desired outcome (accurate predictions).

What is Optuna?

Optuna is an open-source library for hyperparameter optimization. It provides a framework for efficiently searching through a vast space of possible hyperparameter combinations to identify the set that yields the best performance for a given model.

- Optuna will iteratively try different hyperparameter combinations, evaluate the student model's performance using the objective function, and prioritize exploring promising regions of the search space.

- Once Optuna converges, it will provide the hyperparameter combination that yielded the best performance on the validation dataset according to your objective function.

Conclusion

In wrapping up, it's important to appreciate the amazing role that knowledge distillation plays in the world of artificial intelligence. It's like a magic wand that can transfer wisdom from a more complex model to a simpler buddy, without losing a beat in performance. This technique is a superstar, especially when we're working with limited computational resources. With a sprinkle of loss functions and a dash of hyperparameter tuning, we can ensure that our student model soaks up knowledge from the teacher model like a sponge. Plus, with handy tools like Optuna, the task of hyperparameter optimization becomes a breeze, leading to even more powerful student models. As we continue on this exciting journey of exploration and innovation, we can't wait to see further leaps in model efficiency and performance.

> 💡 In this presentation, we will explore the concept of knowledge distillation in AI models. We will discuss what it is, how it works, and the benefits it offers.
