Transfer Learning

Transfer learning is a machine learning (ML) technique where knowledge gained from a previous task is reused to improve performance on a new, related task. Instead of building a new model from scratch for each task, transfer learning leverages pre-trained models as a starting point. This is especially useful in deep learning, where training deep neural networks can require large amounts of data and computational resources.

During transfer learning, a trained machine learning model transfers its knowledge to a different but closely linked problem to improve performance. For example, if you trained a simple classifier to predict whether an image contains a backpack, you could use the model's training knowledge to identify other objects such as sunglasses.

Benefits of Transfer Learning

- **Increases Learning Speed** With fewer new things to learn, the algorithm generates high-quality output faster.
- **Reduces Data Needs** Algorithms can learn even when fed with less training data.
- **Reduces Computational Cost** Transfer learning reduces the costs of building models by enabling them to reuse previously trained parameters.
- **Improved Performance** Performance can improve due to the knowledge it gains from training on the first task.
- **Prevents Overfitting** Transfer learning helps prevent overfitting, especially when target data sets are small.

How it Works

In transfer learning, the weights and parameters learned by a model on a source task are transferred to a new model for a target task. This can involve using the pre-trained model as a feature extractor or fine-tuning the entire model on the new task. The idea is that the knowledge gained from the original task, such as recognizing edges and shapes in images, can be useful for the new task.

Applications

Transfer learning can be applied across various industries:

- **Image Recognition** A model trained to identify dogs can be used to identify cats.
- **Natural Language Processing (NLP)** Knowledge of pre-trained AI models that understand linguistic structures can be transferred to other models.
- **Medical Imaging** Pre-trained models are used as starting points for analyzing medical images, helping to classify tumors or detect abnormalities.
- **Robotics** Robots can adapt skills learned in one environment to new environments.
- **Spam Filtering** Models trained on generic email spam detection can be fine-tuned for specific industries.

What are the steps in transfer learning?

There are three main steps when fine-tuning a machine-learning model for a new task.

Select a pre-trained model

First, select a pre-trained model with prior knowledge or skills for a related task. A useful context for choosing a suitable model is to determine the source task of each model. If you understand the original tasks the model performed, you can find one that more effectively transitions to a new task.

Configure your pre-trained models

After selecting your source model, configure it to pass knowledge to a model to complete the related task. There are two main methods of doing this.

Freeze pre-trained layers

Layers are the building blocks of neural networks. Each layer consists of a set of neurons and performs specific transformations on the input data. Weights are the parameters the network uses for decision-making. Initially set to random values, weights are adjusted during the training process as the model learns from the data.

By freezing the weights of the pre-trained layers, you keep them fixed, preserving the knowledge that the deep learning model obtained from the source task.

Remove the last layer

In some use cases, you can also remove the last layers of the pre-trained model. In most ML architectures, the last layers are task-specific. Removing these final layers helps you reconfigure the model for new task requirements.

Introduce new layers

Introducing new layers on top of your pre-trained model helps you adapt to the specialized nature of the new task. The new layers adapt the model to the nuances and functions of the new requirement.

Train the model for the target domain

You train the model on target task data to develop its standard output to align with the new task. The pre-trained model likely produces different outputs from those desired. After monitoring and evaluating the model's performance during training, you can adjust the hyperparameters or baseline neural network architecture to improve output further. Unlike weights, hyperparameters are not learned from the data. They are pre-set and play a crucial role in determining the efficiency and effectiveness of the training process. For example, you could adjust regularization parameters or the model's learning rates to improve its ability in relation to the target task.