practicum
building ai knowledge

Evaluating and Optimizing Transfer Learning

Transfer Learning Concepts - Objectives

1. Critique pre-trained models for suitability and select candidates.
2. Assess trade-offs between transfer learning and full training.
3. Develop intuition on choosing appropriate TL techniques.
4. Evaluate the outcome of different TL approaches.
5. Troubleshoot common issues like catastrophic forgetting and negative transfer.

Source Model Selection Factor 1: Architecture and Task Fit

Example Architecture Type	Best Suited For	Common Examples
CNNs (Convolutional Neural Networks)	Image classification, object detection	ResNet, VGG, EfficientNet
Transformers (NLP)	Text classification, language generation, question answering	BERT, GPT, T5
Multimodal Models	Image + text understanding, captioning, vision-language models	CLIP, BLIP

Evaluating Pre-Trained Model Suitability

MATCHING SOURCE & TARGET TASKS?

SIMILAR SOURCE & TARGET TRAINING DATA?

HOW LARGE IS MY TARGET DATASET?

WHAT ARE MY COMPUTE RESOURCES (GPU/TPU ACCESS)?

INFERENCE SPEED?

A Note on Model Size

Example Model	Relative Size	Number of Parameters	Relative Speed	Best Use Cases
ResNet-18	Small	~11 million	Fast	Embedded devices, real-time applications
ViT-Large	Medium	~307 million	Medium	High-resolution image tasks
Mistral-Large-2	Large	~123 billion	Slow	Small scale language processing, focused on math and coding.
GPT-4	Very Large	~1.8 trillion	Very Slow	Large-scale general language processing

Quick Quiz!

Question 1: You need to deploy a model on a mobile phone for real-time image classification. Which factor is MOST critical when selecting a pre-trained model?

A) Maximum possible accuracy regardless of size.
B) Model size and inference speed (latency).
C) Whether the model was trained on the exact same dataset.
D) The availability of a TensorFlow version.

Answer: B

The Big Decision: Full Training vs Transfer

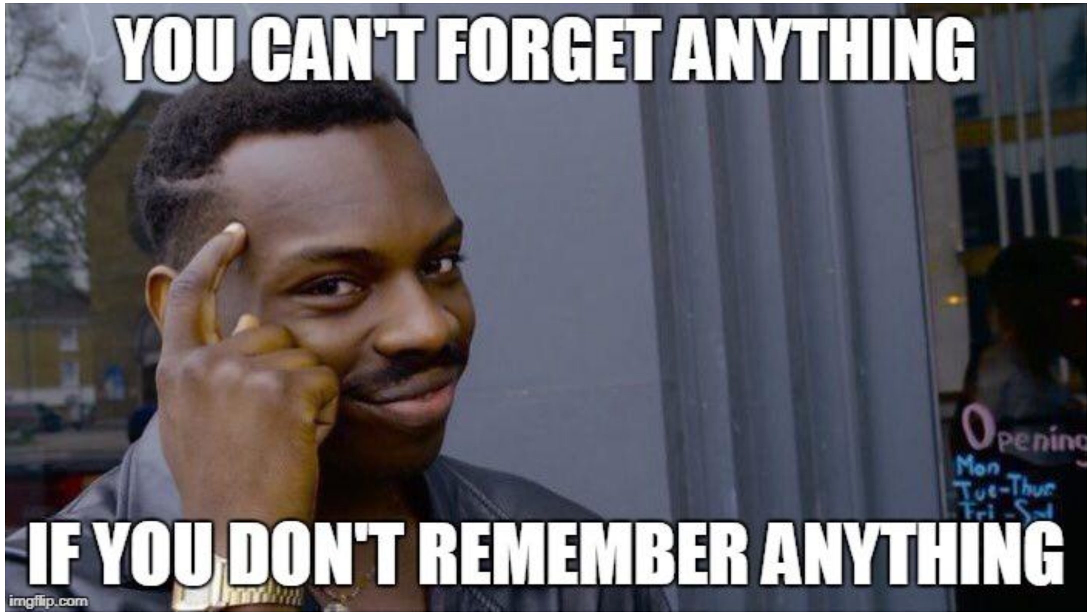
Approach	Transfer Learning	Full Training
Data Requirement	Requires a smaller labeled dataset	Needs a large, labeled dataset
Training Time	Generally faster training	Potentially longer, though highly architecture dependent.
Computational Cost	Less expensive, can run on consumer GPUs	Can require significant GPU resources
Performance	Often achieves high accuracy with limited data	Can outperform transfer learning if sufficient data is available
Flexibility	Limited by pre-trained model's learned features	Fully customizable for the specific task

Comparing Specific TL Techniques

Technique	Best Used When…
Feature Extraction	Data is limited, and the target task is similar to the source task.
Fine-Tuning	The target task is moderately different from the source task, and sufficient data is available.
LoRA (Low-Rank Adaptation)	The target task requires adaptation but computational resources are limited.

Common Pitfalls # 1: Catastrophic Forgetting

- **What:** Model forgets source knowledge when fine-tuned on new task.
- **Why:** Small target dataset + too many layers fine-tuned; learning rate too high; overfitting.
- **Troubleshooting:**
  - Freeze early layers, fine-tune only later ones.
  - Lower learning rate (e.g., 1e-5).
  - Gradually unfreeze layers.
  - Use regularization.



Common Pitfalls # 2: Negative Transfer

- **What:** Transferred knowledge hurts performance on the target task.
- **Why:** Source/target domains too different; wrong layers fine-tuned; source model biases interfere.

- **Troubleshooting:**
  - Feature Extraction instead of Fine-tuning if domains differ greatly.
  - Domain adaptation techniques.
  - Freezing different layers.

Common Pitfalls # 3: Domain Shift

- **What:** Target data distribution differs significantly from source data distribution, causing poor generalization.
- **Why:** Differences in lighting, resolution, backgrounds, classes, input modality (i.e. satellite vs drone).
- **Troubleshooting:**
  - Data Augmentation.
  - Fine-tune on a subset of target data first.
  - Normalize/preprocess target data.

The Role of Regularization

DROPOUT

L2 WEIGHT DECAY

DATA AUGMENTATION

Case Study: Adapting a Sentiment Analysis Model

A company wants to fine-tune **BERT** to classify **customer reviews** as positive, neutral, or negative. However, they encounter **common transfer learning issues** along the way.

- **Issue:** The model loses its ability to understand general sentence structure when fine-tuning on the customer review dataset.

- **Problem?** Catastrophic Forgetting

- **Solution?** Lower the learning rate and freeze early transformer layers.

Case Study: Adapting a Sentiment Analysis Model

A company wants to fine-tune **BERT** to classify **customer reviews** as positive, neutral, or negative. However, they encounter **common transfer learning issues** along the way.

- **Issue:** The pre-trained model was originally trained on Wikipedia and news articles, but the customer reviews contain slang and informal language.
- **Problem?** Negative Transfer
- **Solution?** Use **domain adaptation** by first fine-tuning on a **larger dataset of informal text** before specializing on customer reviews.

Case Study: Adapting a Sentiment Analysis Model

A company wants to fine-tune **BERT** to classify **customer reviews** as positive, neutral, or negative. However, they encounter **common transfer learning issues** along the way.

- **Issue:** Customer reviews often contain emoji-based sentiment that the model struggles with.
- **Problem?** Domain Shift
- **Solution?** Add **emoji tokenization** and fine-tune the model **on augmented data** that includes emojis.

Pre-Trained Model Evaluation Exercise!!!