Learning From Data
Lecture 13: Deep Representation Learning and Foundation Models

Yang Li yangli@sz.tsinghua.edu.cn

TBSI

December 20, 2024

Final Poster Session Information

When & Where

December 27 9:30am-12:00pm @ International Phase I Building A, 1F

What to include in the poster?

- Abstract: a short summary of your work (no more than 100 words)
- Introduction/Motivation: why is this problem important and what is your contribution?
- Method: the machine learning methodology used
- Results: the dataset you use and the experimental results (tables and figures)
- Conclusion/Discussion: conclude your technical/application contributions
- Reference: include 2-3 important references (You can use smaller fonts for this part.)

Each group needs to submit a poster in PDF format of A0-size to Web Learning before **December 25 12:00pm (noon)**.

Today's Lecture

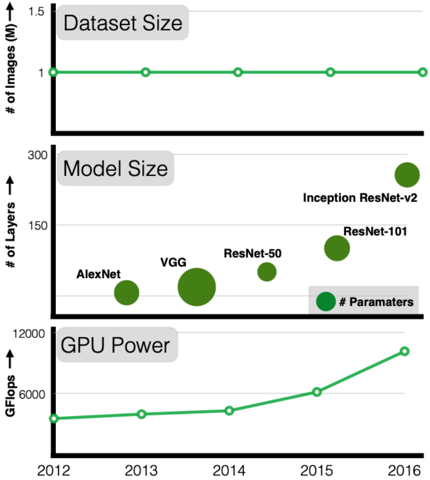
- What is self-supervised representation learning?
- Self-supervised pre-training of foundational models
  - BERT (text representation)
  - MAE (image representation)
  - CLIP (image-text)
- Adapting foundation models to downstream tasks

Introduction&Motivation

The Label Bottleneck in Supervised Learning

- Conventional DNNs has been trained with supervised learning

ImageNet(2012): 1.33Million labeled images

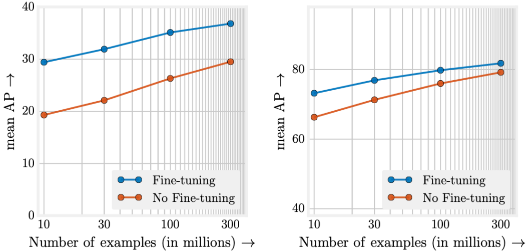


Why didn't vision benchmark data size increase for five years? [1]

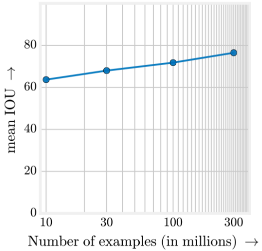
[1] Sun, Chen, et al. "Revisiting unreasonable effectiveness of data in deep learning era." Proceedings of the IEEE International Conference on Computer Vision. 2017.

The Label Bottleneck in Supervised Learning

Performance in supervised vision tasks increases **logarithmically based on the size of (labeled) training data**.



Object detection (COCO minival, PASCAL VOC 2007)



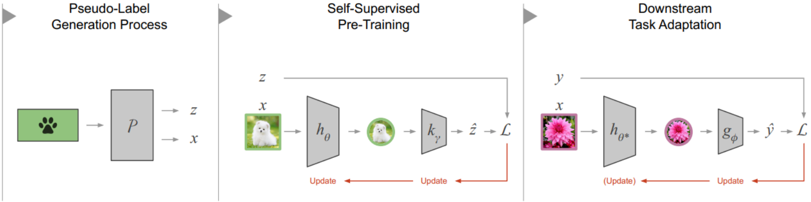
Semantic segmentation (Pascal VOC 2012)

- Annotating data is costly, time-consuming and requires domain expertise.
- Many real-world applications lack sufficient labeled data (e.g., medical imaging, low-resource languages).

Alleviate this bottleneck by learning from abundant ***unlabeled data***.

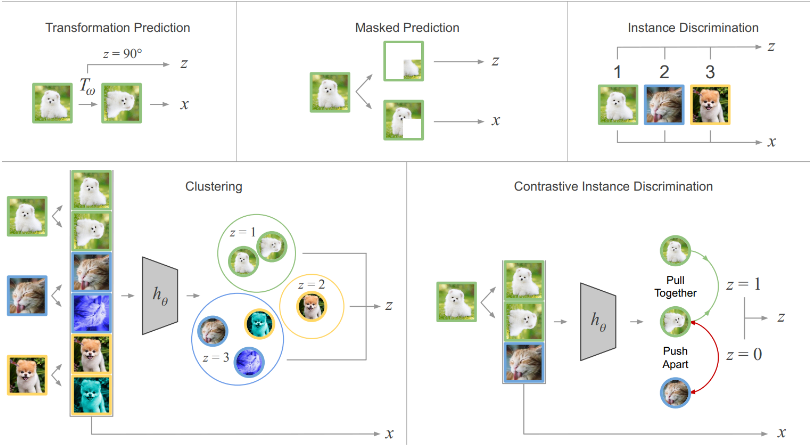
Leveraging Unlabeled Data

- **Unlabeled data** are vast and readily available in all domains:
  - Text data (e.g., web pages, books).
  - Image and video data (e.g., YouTube, Flickr).
  - Audio data (e.g., speech recordings, podcasts).
- **Self-Supervised Representation Learning (SSRL):**
  - Extracts rich features from unlabeled data using **pretext tasks**.
  - Learned representations are **universal** and can be reused for various downstream tasks (transfer learning).



How to Design Pre-text Task

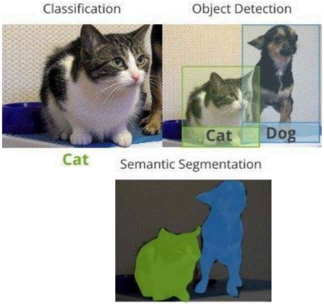
- **Four families of pre-text tasks**: transformation prediction, masked prediction, instance discrimination and clustering
- Constrastive SSL is a special case of instance discrimination



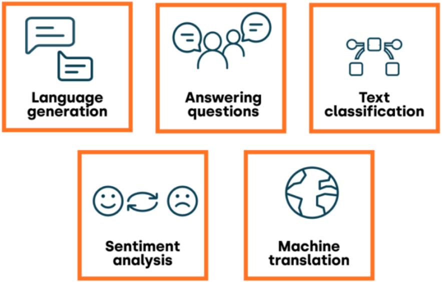
Transfer Learning and Knowledge Reuse

- Many vision/NLP tasks that can benefit from pre-trained SSRL models by knowledge transfer

Vision Tasks



NLP Tasks



- A pre-trained model with **good representations** would
  - Give better initialization for downstream tasks with limited labels.
  - Have faster data efficiency and faster convergence.
  - Generalizes better to unseen tasks and domains.

Example: Self-supervised representation learning in BERT

From Context-Independent to Context-Sensitive in NLP

- **Context-Independent Representations:**
  - Models like word2vec and GloVe assign the same vector to a word regardless of context.
  - Limitation: Cannot capture polysemy; e.g., "bank" in "river bank" vs. "financial bank".
- **Context-Sensitive Representations:**
  - Models such as ELMo generate word representations that vary with context.
  - Achieved by processing entire sequences and capturing contextual nuances.

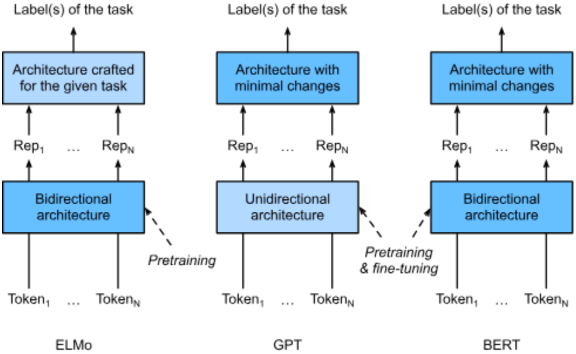
From Task-Specific to Task-Agnostic in NLP

- **Task-Specific Architectures:**
  - ELMo integrates with models tailored for specific NLP tasks.
  - Requires designing unique architectures for each task.
- **Task-Agnostic Models:**
  - GPT employs a general architecture applicable across various tasks.
  - Utilizes a Transformer decoder with unidirectional (left-to-right) context encoding.
  - Limitation: May not fully capture context for words influenced by right-side context.

BERT: Bidirectional Encoder Representations from Transformers

Pretraining Tasks:

- **Masked Language Model (MLM):** Predict masked tokens using bidirectional context.
- **Next Sentence Prediction (NSP):** Learn relationships between sentences.



Devlin, Jacob. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

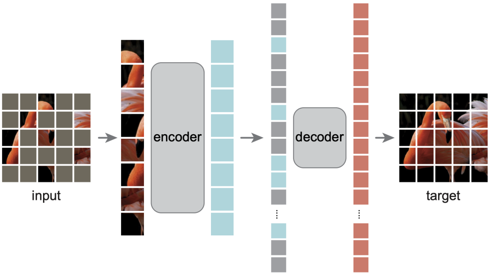
Masked Language Modeling (MLM) in BERT

- **Objective:**
  - Enable bidirectional context understanding by predicting randomly masked tokens within a sequence.
- **Process:Model predicts the original token for each masked position.**
  - a special "<mask>" token for 80% of the time (e.g., "this movie is great" becomes "this movie is <mask>");
  - a random token for 10% of the time (e.g., "this movie is great" becomes "this movie is drink");
  - the unchanged label token for 10% of the time (e.g., "this movie is great" becomes "this movie is great").
- **Benefits:**
  - Allows BERT to capture context from both left and right, enhancing understanding of word meaning based on surrounding words.

Example: Self-supervised representation learning in Masked Autoencoder

MAE:Masked Autoencoders Are Scalable Vision Learners

- Learns to reconstruct missing (masked) portions of input images using an encoder-decoder architecture.
- **Architecture:**
  - **Encoder:** Operates on a subset of visible patches, making it lightweight and efficient.
  - **Decoder:** Reconstructs the full image from encoded representations, including the masked patches.
  - **Training:** End-to-end training with MSE loss.



He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

Why Use MAE?

- **Advantages of MAE:**
  - **Efficient Training:** Processes only a subset of image patches, reducing computation.
  - **Strong Representation:** Learns meaningful representations from unannotated data.
  - **Scalability:** Performs well on large-scale datasets without extensive fine-tuning.
  - **Performance:** Outperforms supervised models in downstream tasks like image classification.

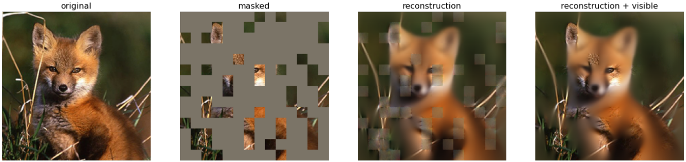


Illustration of MAE results.

Example: Self-supervised representation learning in CLIP

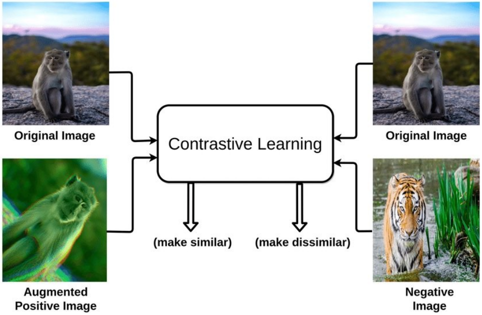
Motivation for CLIP

- **Goal of CLIP:** Learn a **joint representation** of images and text using:
  - Unlabeled image-text pairs freely available on the internet.
  - Contrastive learning to align visual and textual features.
- Enables zero-shot transfer to downstream tasks without additional training.

Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.

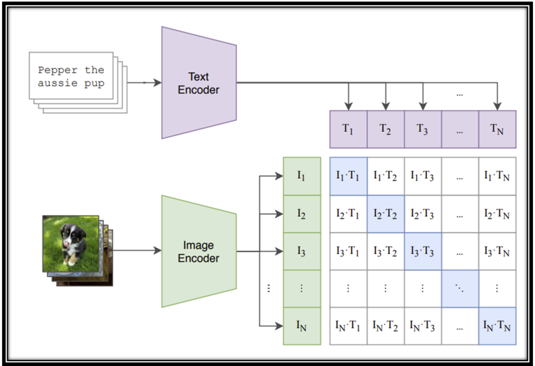
What is Contrastive Language-Image Pre-Training (CLIP)?

- Developed by OpenAI to connect visual and textual data in a shared embedding space.
- **How it works:**
  - Given a set of images and corresponding textual descriptions, CLIP learns to:
    - Align images and their correct textual descriptions (positive pairs).
    - Discriminate between unrelated images and texts (negative pairs).



CLIP Architecture

- CLIP consists of two main components:
  1. **Image Encoder:**
     - Typically a Vision Transformer (ViT) or ResNet.
     - Maps an input image into a feature vector.
  2. **Text Encoder:**
     - A Transformer-based model similar to GPT or BERT.
     - Converts input text into a feature vector.
- The image and text feature vectors are projected into a **shared embedding space**.



CLIP Pre-training Strategy

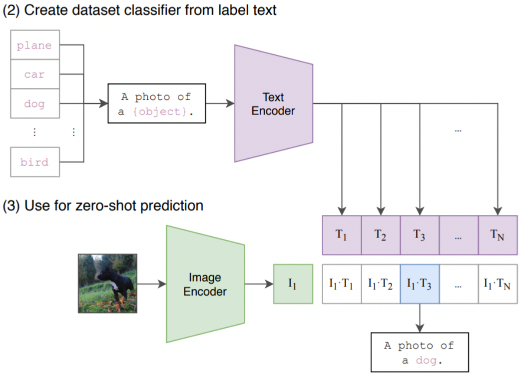
- CLIP is pre-trained using **massive datasets** of image-text pairs scraped from the Internet.
- **Contrastive Learning:** Minimize contrastive loss:

$$\text{Loss} = -\log \frac{\exp(\text{similarity}(\text{image}_i, \text{text}_i))}{\sum_j \exp(\text{similarity}(\text{image}_i, \text{text}_j))}$$

  - Positive pairs are pulled closer.
  - Negative pairs are pushed apart.
- **Scalable Learning:**
  - The large scale of the data enables generalization to unseen tasks.
  - CLIP learns **the semantic relationships** between vision and language.

Use CLIP for Zero-shot Image Classification

1. **Prepare & Encode the Inputs:**
   - Pass the image through the image encoder
   - Define a set of text prompts describing possible classes and encode each text prompt using the text encoder to obtain feature vectors.
2. **Compute Similarity:** Measure the cosine similarity between the image feature vector and each text feature vector.
3. **Classify the Image:** Assign the class label corresponding to the text prompt with the highest similarity score.



Advantages of CLIP

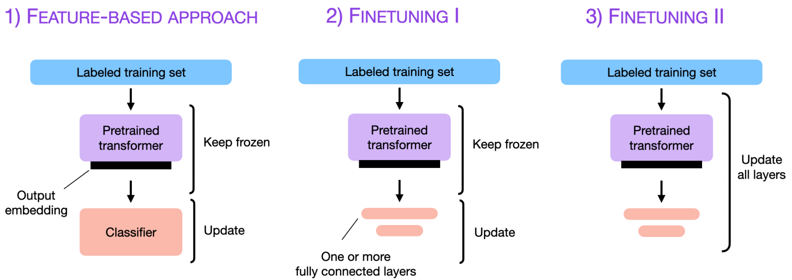
- **Zero-Shot Learning:**
  - CLIP can perform tasks without additional training on labeled data.
  - Example: Classifying objects in an image based on text prompts (e.g., "a dog," "a cat").
- **Task-Agnostic:**
  - CLIP's learned representations can generalize to diverse tasks, including:
    - Image classification.
    - Object detection.
    - Image retrieval.
- **Reduced Label Dependency:**
  - Eliminates the need for manually annotated datasets.
  - Uses natural image-text pairs freely available online.
- **Scalability:** Performance improves as more image-text data becomes available.

Applications of CLIP

- **Zero-Shot Image Classification:**
  - Example: Classify objects in an image using text prompts like "a car" or "a plane."
- **Image Retrieval:**
  - Retrieve images that match a text query (e.g., "a sunny beach").
- **Visual Question Answering (VQA):**
  - Combine images and text to answer natural language questions about visual content.
- **Content Moderation:**
  - Automatically detect inappropriate or harmful content in images based on textual prompts.
- **Multi-Modal Applications:**
  - Connect vision and language for robotics, autonomous vehicles, and AI assistants.

How to adapt pre-trained model?

Supervised Finetuning

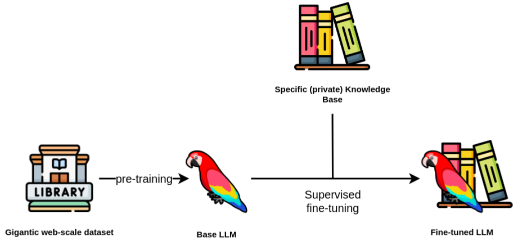


Which layer can be transferred (copied)?

- Speech: usually copy the last few layers
- Image: usually copy the first few layers

Motivation for SFT

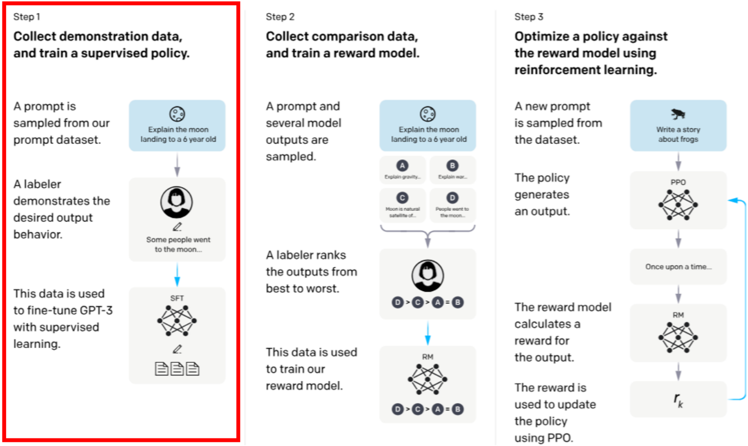
- **Adapting Pre-trained Models to Specific Tasks.** Pre-trained models require supervised fine-tuning (SFT) to adapt their general features to the specific requirements of a target task.
- **Reducing Training Data and Resources.** SFT leverages the knowledge gained during pre-training, reducing the need for large labeled datasets and computational resources.
- **Improving Model Performance on Complex Tasks.** As models become larger and more complex, supervised fine-tuning allows them to better capture and optimize task-specific features, significantly improving performance on challenging or specialized tasks.



SFT in chatGPT

The training consists of three steps:

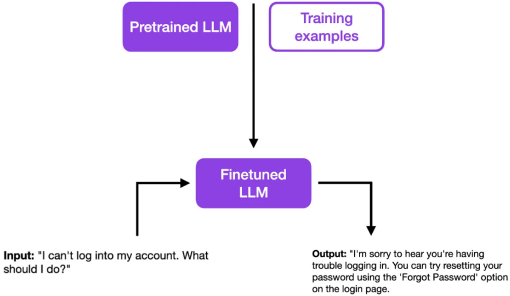
- **Supervised fine-tuning (SFT)**
- Reward model (RM) training
- Reinforcement learning via proximal policy optimization (PPO)



Ouyang, Long, et al. "Training language models to follow instructions with human feedback." Advances in neural information processing systems 35 (2022): 27730-27744.

SFT in chatGPT

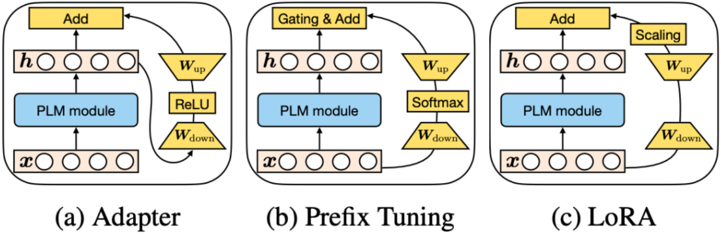
- **Data Preparation**
  SFT requires a labeled dataset specific to the target task, such as question-answer pairs.
- **Fine-Tuning Process**
  The pre-trained model is further trained on this labeled data using supervised learning. (e.g. GPT-3)
- **Task-Specific Optimization**
  Through SFT, the model's general knowledge is refined to handle specialized tasks more effectively.



Parameter-Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning

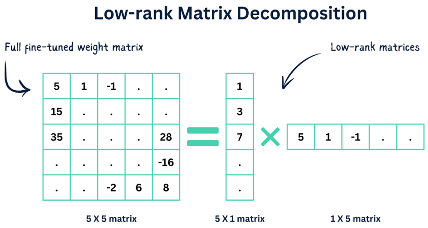
- PEFT adjusts only a *small subset of parameters* in a pre-trained LLM, freezing the original weights and adding a few new parameters to fine-tune on a task-specific dataset.
- Advantages:
  - Reduces computational cost
  - Quick adaptation to new tasks with limited data
  - Scalable for large models and multiple tasks



He, Junxian, et al. "Towards a unified view of parameter-efficient transfer learning." arXiv preprint arXiv:2110.04366 (2021).

LoRA (Low-Rank Adaptation)

- **Traditional Fine-Tuning**: Modifies the pre-trained network's weight matrix as $W' = W + \Delta W$.
- **Intrinsic Rank Hypothesis**: LoRA introduces the hypothesis that not all weight updates are equally important, only a small subset of ΔW contributes significantly to the model's performance for the new task.

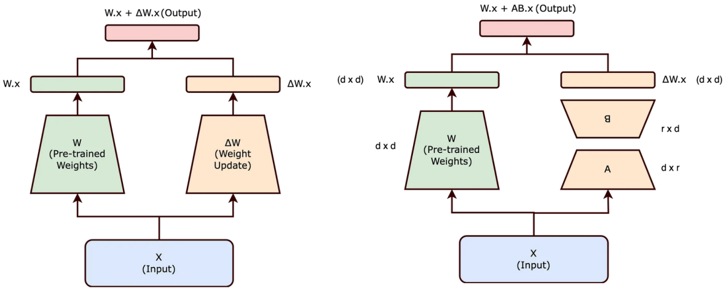


Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021).

LoRA (Low-Rank Adaptation)

- **Matrix Decomposition**:By decomposing the update into two smaller matrices, LoRA reduces the number of parameters to be learned. The new weights can then be expressed as:

$$W' = W + BA$$



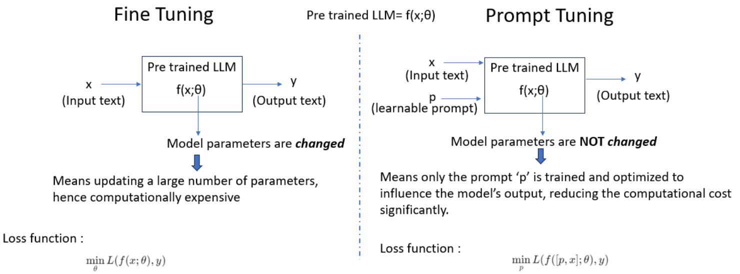
LoRA (Low-Rank Adaptation)

Advantages of LoRA

- **Parameter Efficiency:** Only a small number of low-rank parameters are optimized, making LoRA much more parameter-efficient compared to traditional fine-tuning methods.
- **Reduced Computational Costs:** With fewer parameters to update, LoRA requires less computation.
- **Flexibility:** LoRA can be applied to various types of pre-trained models.

Prompt Tuning

- Prompt tuning is a method of fine-tuning a language model by **optimizing the input prompt** (rather than adjusting the model's parameters). It modifies the prompt to guide the model's output toward the desired behavior.
- Learnable input or soft prompts are added to the input text.
- The input prompt is modified so that the model produces a desired output.

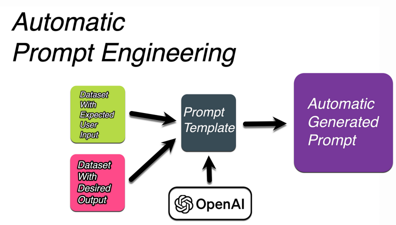


Lester, B., Al-Rfou, R. and Constant, N., 2021. The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

Prompt Tuning

Applications of Prompt Tuning

- **Text Generation:** Adjusting prompts to control style, tone, or content in tasks such as writing, story generation, or dialogue systems.
- **Text Classification:** Using prompts for different categories, optimizing the model for specific classification tasks.
- **Machine Translation:** Fine-tuning translation tasks by adjusting the prompt to ensure better and contextually accurate translations.



Additional Readings

- **Survey on self-supervised representation learning**: Ericsson, Linus, et al. "Self-supervised representation learning: Introduction, Advances, and Challenges." IEEE Signal Processing Magazine 39.3 (2022): 42-62.
- **Natural Language Processing: Pretraining**: `https://d2l.ai/chapter_natural-language-processing-pretraining/index.html`
- **Survey on parameter efficient fine tuning**: Ding, N., Qin, Y., Yang, G. et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nat Mach Intell 5, 220–235 (2023).