Week 6: Pre-trained Language Models

Pre-Class Discovery & Post-Class Application

How to Use This Handout

- **Before Class**: Complete Part A to discover why we need pre-training
- **During Class**: Learn about BERT, GPT, and fine-tuning techniques
- **After Class**: Complete Part B to apply technical concepts

---

PART A: PRE-CLASS DISCOVERY

No prior knowledge required - Let's discover transfer learning!

1 A1: The Learning Problem (10 minutes)

1.1 Learning from Scratch vs Building on Knowledge

Real World

Imagine you want to learn to play the piano. Which approach would be faster?

1. Starting from zero: Learn what music is, what notes are, how to read music, finger positions, then finally play songs
2. Already knowing music theory: Just learn finger positions and practice

Exercise: Think about learning a new skill. List what you could reuse from existing knowledge:

New skill: Learning Spanish

- Can reuse from English: _____
- Must learn fresh: _____
- Time saved by reusing: _____

Q: If an AI needs to analyze movie reviews, what language skills would it need?

List 5 skills:

1. _____
2. _____
3. _____
4. _____

5. _____

Think: Should every company teach their AI what "the" means from scratch?

Discovery

You've discovered the motivation for pre-training: Most language understanding is general and can be learned once and reused!

2 A2: The Fill-in-the-Blank Game (10 minutes)

2.1 How We Understand Context

Exercise: Fill in the blanks using context clues:

1. "The cat sat on the _____"
2. "I went to the _____ to buy milk"
3. "The _____ was delicious but too expensive"
4. "She _____ the ball to her friend"

Q: Which clues did you use to fill each blank?

For sentence 2:

- Words before the blank: _____
- Words after the blank: _____
- Your world knowledge: _____

2.2 One-Way vs Two-Way Understanding

Intuition

Imagine reading a mystery novel:

- Reading forward only: You guess what happens next
- Reading the whole page: You understand what's happening

Which gives better understanding?

Exercise: Compare these two approaches for filling blanks:

"The [BLANK] barked loudly at the mailman"
Approach 1: Only see "The [BLANK]" Possible answers: _____
Approach 2: See entire sentence Possible answers: _____
Which approach is more accurate? Why?

3 A3: The Prediction Game (10 minutes)

3.1 Completing Sentences

Exercise: Complete these sentence beginnings:

1. "Once upon a _____"
2. "The weather today is _____"
3. "I love to eat _____"
4. "The best thing about weekends is _____"

Q: How did you predict what comes next?

Think about your process:

- Did you use grammar rules? _____
- Did you use common patterns? _____
- Did you use personal experience? _____

3.2 Generation vs Understanding

Think: What's the difference between:

1. Filling a blank in the middle of a sentence
2. Continuing a sentence from where it stops

Which requires more creativity? _____ Which requires more understanding? _____

Discovery

You've discovered two approaches: BERT-style (fill blanks using all context) and GPT-style (predict what comes next)!

4 A4: The Specialization Problem (10 minutes)

4.1 General vs Specific Knowledge

Real World

A medical student knows:

- General: How to read, write, study
- Specific: Medical terminology, procedures

The general knowledge transfers to any field!

Exercise: Categorize these AI tasks as General or Specific:

Task	General/Specific
Understanding grammar	_____
Knowing movie genres	_____
Recognizing sentiment words	_____
Understanding "the" and "a"	_____
Medical diagnosis terms	_____
Sentence structure	_____

Q: What percentage of language understanding is general vs task-specific?

Your estimate: ___% general, ___% specific

5 A5: The Cost Problem (5 minutes)

5.1 Training Efficiency

Exercise: Calculate the waste:

5 companies each need sentiment analysis AI:

- Each trains from scratch: $500,000
- Total cost: _____

Alternative approach:

- Train one general model: $1,000,000
- Each company adapts it: $10,000
- Total cost: _____
- Money saved: _____

Think: Besides money, what else is wasted when training from scratch?

Checkpoint

Pre-Class Complete! You've discovered:

- Why reusing knowledge is efficient
- Two ways to learn from text (fill blanks vs predict next)
- General vs specific knowledge
- The economic motivation for pre-training

Ready to learn how BERT and GPT implement these ideas!

PART B: POST-CLASS APPLICATION

Now apply the technical concepts from lecture!

6 B1: BERT's Masked Language Modeling (15 minutes)

6.1 Understanding MLM

Technical Detail

BERT's training objective:

- Randomly mask 15% of tokens
- 80% replaced with [MASK]
- 10% replaced with random token
- 10% kept unchanged

This prevents the model from only looking for [MASK].

Exercise: Implement masking strategy:

Given sentence: "The quick brown fox jumps" Token positions: [0, 1, 2, 3, 4]
If we mask 15% (1 token), and choose position 2 ("brown"):

- 80% chance: "The quick [MASK] fox jumps"
- 10% chance: "The quick [RANDOM] fox jumps"
- 10% chance: "The quick brown fox jumps"

Q: Why not mask 50% of tokens? What would happen?

6.2 Bidirectional Attention

Exercise: Calculate attention patterns:

For "The [MASK] sat on the mat", BERT can attend:

From [MASK] to:	The	[MASK]	sat	on	the	mat
Can attend?	Yes	Yes	Yes	Yes	Yes	Yes

For GPT predicting after "cat":

From position 2 to:	The	cat	sat	on	the	mat
Can attend?	Yes	Yes	No	No	No	No

Q: How many connections does BERT have vs GPT for a sequence of length n?

BERT: _____ connections GPT: _____ connections

7 B2: GPT's Autoregressive Modeling (10 minutes)

7.1 Next Token Prediction

Technical Detail

GPT's training objective:

$$P(x_t|x_1, x_2, ..., x_{t-1}) = \text{softmax}(W_o \cdot h_t)$$

where h_t is the hidden state at position t.

Exercise: Calculate probabilities:

Given context: "The cat" Vocabulary: ["sat", "ran", "jumped", "slept"] Logits: [2.5, 1.0, 0.5, 1.5] After softmax:

$$P(\text{sat}) = \frac{e^{2.5}}{e^{2.5} + e^{1.0} + e^{0.5} + e^{1.5}} = \text{\_\_\_\_} \quad (1)$$

$$P(\text{ran}) = \text{\_\_\_\_} \quad (2)$$

$$P(\text{jumped}) = \text{\_\_\_\_} \quad (3)$$

$$P(\text{slept}) = \text{\_\_\_\_} \quad (4)$$

7.2 Generation Strategy

Q: What happens if we always pick the highest probability word?

Q: How does temperature affect generation?

Temperature = 0.5: _____ Temperature = 1.0: _____ Temperature = 2.0: _____

8 B3: Fine-tuning Mathematics (10 minutes)

8.1 Transfer Learning

Technical Detail

Fine-tuning updates:

$$\theta_{new} = \theta_{pretrained} + \alpha \cdot \nabla_\theta L_{task}$$

where α is the learning rate (typically small: 2e-5)

Exercise: Compare learning rates:

Training Type	Typical LR	Why?
From scratch	1e-3	Need large updates
Fine-tuning	2e-5	_____

Q: What happens with too large learning rate during fine-tuning?

8.2 Catastrophic Forgetting

Exercise: Design a fine-tuning schedule to prevent forgetting:

1. Initial LR: _____
2. Warmup steps: _____
3. Decay strategy: _____
4. Freeze layers? _____

9 B4: Model Selection (10 minutes)

9.1 Comparing Architectures

Exercise: Fill in the comparison table:

Criterion	BERT	GPT	T5
Parameters	340M	1.5B	11B
Best for	_____	Generation	_____
Training objective	MLM	_____	Span corruption
Architecture	Encoder	_____	Encoder-Decoder
Context	Bidirectional	_____	_____

9.2 Task Matching

Exercise: Match models to tasks:

Task	Best Model
Sentiment analysis	_____
Text generation	_____
Question answering	_____
Translation	_____
Summarization	_____
Named entity recognition	_____

10 B5: Implementation Exercise (15 minutes)

10.1 Fine-tuning BERT

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer, Trainer
import torch

# Load pre-trained model
model = BertForSequenceClassification.from_pretrained(
    'bert-base-uncased',
    num_labels=2  # Binary classification
)

# Freeze embeddings (optional)
for param in model.bert.embeddings.parameters():
    param.requires_grad = # YOUR CODE: True or False?

# Set fine-tuning parameters
training_args = TrainingArguments(
```

```
    output_dir='./results',
    learning_rate=# YOUR CODE: appropriate LR,
    per_device_train_batch_size=# YOUR CODE: batch size,
    num_train_epochs=# YOUR CODE: epochs,
    warmup_steps=# YOUR CODE: warmup,
    weight_decay=# YOUR CODE: decay,
)

# What happens if we set learning_rate=1e-2?
# YOUR ANSWER: _____
```

10.2 Using GPT for Generation

```
from transformers import GPT2LMHeadModel, GPT2Tokenizer

# Load model
model = GPT2LMHeadModel.from_pretrained('gpt2')
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')

# Encode prompt
prompt = "Once upon a time"
inputs = tokenizer.encode(prompt, return_tensors='pt')

# Generate with different strategies
# Greedy decoding
output_greedy = model.generate(
    inputs,
    max_length=50,
    # YOUR CODE: what parameter for greedy?
)

# Sampling with temperature
output_sample = model.generate(
    inputs,
    max_length=50,
    do_sample=True,
    temperature=# YOUR CODE: temperature value
)

# Beam search
output_beam = model.generate(
    inputs,
    max_length=50,
    num_beams=# YOUR CODE: beam size
)
```

11 B6: Advanced Concepts (10 minutes)

11.1 Efficient Fine-tuning

Technical Detail

Parameter-Efficient Fine-Tuning (PEFT):

- LoRA: Low-rank adaptation
- Prefix tuning: Learn soft prompts
- Adapter layers: Small trainable modules

Exercise: Calculate parameter savings with LoRA:

Original model: 340M parameters LoRA rank r=16:

- Original weight: $W \in \mathbb{R}^{768\times768}$
- LoRA: $W + BA$ where $B \in \mathbb{R}^{768\times16}$, $A \in \mathbb{R}^{16\times768}$
- Trainable parameters: _____
- Percentage of original: ____%

11.2 Environmental Impact

Exercise: Calculate carbon footprint:

Training GPT-3:

- Energy: 1,287 MWh
- CO2: 552 tons
- Cost: $4.6M

Fine-tuning GPT-3 on your task:

- Energy: 10 MWh
- CO2: ___ tons
- Cost: _____

Savings by fine-tuning vs training from scratch: ___%

12 B7: Critical Thinking (5 minutes)

Q: What are the limitations of pre-trained models?

List 3 limitations:

1. _____
2. _____
3. _____

Q: How might pre-training bias affect downstream tasks?

Key Takeaways

Checkpoint

After completing both parts, you understand:
From Pre-Class:

- Why transfer learning is powerful
- Intuition for masked modeling vs autoregressive
- General vs task-specific knowledge
- Economic motivation for pre-training

From Post-Class:

- BERT's MLM objective and bidirectional attention
- GPT's autoregressive generation
- Fine-tuning strategies and learning rates
- Model selection criteria
- Implementation with Transformers library

Next Steps

1. Fine-tune BERT on a custom dataset
2. Experiment with different generation strategies in GPT
3. Try parameter-efficient fine-tuning methods
4. Explore domain-specific pre-trained models

Real World

You now understand the technology powering ChatGPT, GitHub Copilot, and modern search engines. These concepts you've learned are worth billions in industry!