Transfer Learning - Comprehensive Question Bank

CSCI316: Big Data Mining Techniques and Implementation

Question 1: Foundational Concepts and Theory (8 marks)

Part A (3 marks)

Explain the "data hunger problem" in deep learning and why it poses particular challenges for small companies. Provide specific examples from the computer vision domain to support your answer.

Part B (5 marks)

Define transfer learning and describe the three key components of the transfer learning framework. Illustrate your answer with a diagram showing the relationship between teacher model, student model, and knowledge transfer process.

Question 2: Technical Architecture Analysis (10 marks)

Part A (4 marks)

Given a 12-layer deep neural network serving as a teacher model:

- How many layers can typically be transferred directly? Show your calculation using the formula K = N-1.
- Explain the layer hierarchy principle and why lower layers are more transferable than higher layers.

Part B (6 marks)

Compare and contrast the architecture of teacher and student models in transfer learning. Draw detailed diagrams showing:

- Teacher model structure with layer breakdown
- Student model structure after transfer
- Identify which components are retained and which are replaced

---

Question 3: Practical Implementation - Face Recognition Case Study (12 marks)

Part A (4 marks)

Analyze the VGG-Face case study provided in the lecture materials:

- Calculate the total number of training images available for the teacher model

- Calculate the total number of training images available for the student model
- Determine the data reduction ratio between teacher and student datasets

Part B (4 marks)

Manual Calculation: Given the performance results from the face recognition case study:

- Without Transfer Learning: 1% accuracy
- With Transfer Learning: 93.47% accuracy

Calculate:

1. The absolute accuracy improvement
2. The relative improvement percentage
3. The error rate reduction achieved through transfer learning

Part C (4 marks)

Explain why the student model achieved such dramatic performance improvement despite having significantly less training data. Discuss the role of feature reusability and provide technical justification for the 15-layer transfer strategy.

Question 4: Coding Implementation - From Scratch (10 marks)

Part A (6 marks)

Code Implementation: Write a Python function from scratch that implements the basic transfer learning process. Your function should:

```python
def transfer_learning_model(teacher_weights, student_layers, freeze_layers=True):
    """
    Implement basic transfer learning without using pre-built frameworks

    Parameters:
    teacher_weights: Dictionary containing pre-trained weights
    student_layers: Number of layers for student model
    freeze_layers: Boolean to freeze transferred layers

    Returns:
    student_model: Configured student model ready for training
    """
    # Your implementation here
    pass
```

Include error handling and comments explaining each step.

Part B (4 marks)

Write a function to calculate the similarity between teacher and student domains:

```python
def domain_similarity_score(teacher_features, student_features):
    """
    Calculate domain similarity score for transfer learning viability

    Parameters:
    teacher_features: Feature vectors from teacher domain
    student_features: Feature vectors from student domain

    Returns:
    similarity_score: Float value between 0 and 1
    """
    # Implement cosine similarity or correlation-based approach
    pass
```

Question 5: Big Data Integration - Spark and TensorFlow (8 marks)

Part A (4 marks)

Design a transfer learning pipeline architecture using Apache Spark and TensorFlow. Your design should show:

- Data flow from raw input to deployed model
- Role of Spark in data processing
- Role of TensorFlow in model training
- Integration points between the two frameworks

Part B (4 marks)

PySpark Code: Write PySpark code to preprocess image data for transfer learning:

```python
```

```
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler

def preprocess_image_data(spark, image_path, target_size=(224, 224)):
    """
    Preprocess images for transfer learning using PySpark

    Parameters:
    spark: SparkSession object
    image_path: Path to image directory
    target_size: Tuple for image resizing

    Returns:
    processed_df: Spark DataFrame with processed images
    """
    # Your PySpark implementation here
    pass
```

Question 6: Advanced Analysis and Real-World Application (12 marks)

Part A (4 marks)

A retail company wants to implement product recognition for their mobile app. They have access to 500 product images per category for 20 categories, but need to recognize 100 categories total. Analyze:

- Whether transfer learning is appropriate for this scenario
- What type of teacher model would be most suitable
- Potential challenges and mitigation strategies

Part B (4 marks)

Manual ROC Analysis: Given the following confusion matrix results from a transfer learning model for medical image classification:

```
Actual vs Predicted
        Disease  Healthy
Disease     85      15
Healthy     10      90
```

Calculate:

1. Sensitivity (True Positive Rate)
2. Specificity (True Negative Rate)

3. Precision and Recall
4. F1-Score

Show all calculation steps.

Part C (4 marks)

Discuss the ethical implications and best practices when using transfer learning in sensitive domains like healthcare or criminal justice. Include considerations for:

- Bias transfer from teacher to student models
- Data privacy and security
- Model interpretability requirements
- Validation and testing protocols

Question 7: Performance Optimization and Fine-tuning (8 marks)

Part A (4 marks)

Explain three different fine-tuning strategies for transfer learning:

1. Freezing all transferred layers
2. Fine-tuning with lower learning rates
3. Gradual unfreezing from top layers

For each strategy, provide the mathematical reasoning and suitable use cases.

Part B (4 marks)

TensorFlow Implementation: Write TensorFlow/Keras code to implement transfer learning with fine-tuning:

```python

```

```
import tensorflow as tf
from tensorflow.keras.applications import VGG16

def create_transfer_model(num_classes, input_shape=(224, 224, 3),
                         fine_tune_layers=0):
    """
    Create transfer learning model with optional fine-tuning

    Parameters:
    num_classes: Number of output classes
    input_shape: Input image dimensions
    fine_tune_layers: Number of layers to fine-tune (0 = freeze all)

    Returns:
    model: Compiled transfer learning model
    """
    # Your TensorFlow implementation here
    pass
```

Question 8: Comparative Analysis and Future Directions (10 marks)

Part A (5 marks)

Compare transfer learning with other machine learning approaches for scenarios with limited training data. Create a table comparing:

- Training time requirements
- Data efficiency
- Computational resources needed
- Performance expectations
- Implementation complexity

Include at least 4 different approaches (e.g., transfer learning, data augmentation, synthetic data generation, few-shot learning).

Part B (5 marks)

Discuss three emerging trends in transfer learning mentioned in the course materials:

1. Multi-task Transfer Learning
2. Domain Adaptation
3. Meta-Learning

For each trend, explain:

- Core concept and methodology
- Advantages over traditional transfer learning
- Current research challenges
- Potential industry applications

Question 9: Problem-Solving Scenario (12 marks)

Scenario:

A startup company developing an autonomous drone delivery system needs to implement object detection for package handling. They have:

- 2,000 images of various packages in controlled environments
- Need to detect 15 different package types
- Limited computational resources
- 3-month development timeline

Part A (4 marks)

Justify why transfer learning is the most appropriate approach for this scenario. Address:

- Data constraints analysis
- Resource limitations
- Timeline pressures
- Domain similarity considerations

Part B (4 marks)

Design a complete transfer learning solution including:

- Teacher model selection with justification
- Architecture adaptation strategy
- Training approach and validation methodology
- Deployment considerations for edge devices

Part C (4 marks)

Risk Assessment: Identify and analyze potential risks in your proposed solution:

- Technical risks and mitigation strategies
- Performance risks and monitoring approaches
- Operational risks and contingency plans

Provide quantitative estimates where possible (e.g., expected accuracy ranges, training time estimates).

Question 10: Integration and System Design (8 marks)

Part A (4 marks)

Design a scalable transfer learning system architecture for a cloud-based image classification service that must handle:

- Multiple concurrent training requests
- Various domain adaptation requirements
- Real-time inference capabilities
- Model versioning and rollback functionality

Part B (4 marks)

Code Integration: Write a Python class that manages the complete transfer learning workflow:

```python
```

```
class TransferLearningManager:
    def __init__(self, teacher_model_path, spark_config=None):
        """
        Initialize transfer learning manager
        """
        pass

    def prepare_data(self, data_path, validation_split=0.2):
        """
        Data preparation using Spark
        """
        pass

    def create_student_model(self, num_classes, fine_tune_layers=0):
        """
        Create and configure student model
        """
        pass

    def train_model(self, epochs=10, batch_size=32):
        """
        Execute training with monitoring
        """
        pass

    def evaluate_performance(self, test_data):
        """
        Comprehensive model evaluation
        """
        pass
```

Include proper error handling and logging mechanisms.

Marking Rubric Guidelines

Theory Questions (Questions 1, 2, 6C, 8):

- **Excellent (90-100%)**: Comprehensive understanding with clear explanations and relevant examples
- **Good (75-89%)**: Good understanding with minor gaps in explanation or examples
- **Satisfactory (60-74%)**: Basic understanding with some conceptual errors
- **Needs Improvement (<60%)**: Limited understanding with significant gaps

Practical Coding (Questions 4, 5B, 7B, 10B):

- **Excellent (90-100%)**: Complete, efficient, well-commented code with error handling
- **Good (75-89%)**: Functional code with minor issues or missing features
- **Satisfactory (60-74%)**: Basic working code with some logical errors
- **Needs Improvement (<60%)**: Non-functional or seriously flawed code

Manual Calculations (Questions 3B, 6B):

- **Excellent (90-100%)**: All calculations correct with clear step-by-step working
- **Good (75-89%)**: Mostly correct with clear methodology, minor computational errors
- **Satisfactory (60-74%)**: Correct approach with some calculation mistakes
- **Needs Improvement (<60%)**: Incorrect methodology or major calculation errors

Applied Analysis (Questions 3, 6A, 9):

- **Excellent (90-100%)**: Thorough analysis with practical insights and realistic solutions
- **Good (75-89%)**: Good analysis with some practical considerations
- **Satisfactory (60-74%)**: Basic analysis with limited practical application
- **Needs Improvement (<60%)**: Superficial analysis with impractical solutions