CHAPTER 6

Speech Processing and Recognition

Summary

1. Introduction and Fundamentals
2. Core Architectures for Speech Recognition
3. Modern Automatic Speech Recognition (ASR) Systems
4. Practical Considerations and Future Directions

1. Introduction and Fundamentals

- A Brief History of Speech Recognition
- The Deep Learning Revolution
- Deep Learning's impact on speech recognition
- Current applications and industry trends
- Key challenges in speech recognition with DNN

A Brief History of Speech Recognition

- **1952: Bell Labs' "Audrey"** - Recognized single digits spoken by a single voice.
- **1960s - 1970s: Hidden Markov Models (HMMs)** - Became the dominant approach, using statistical models to represent speech sounds.
- **1980s - 1990s: Statistical Language Models** - Integration of language models to improve accuracy by considering word probabilities.
- **2000s: Gaussian Mixture Models (GMMs) with HMMs** - Enhanced acoustic modeling using GMMs to represent the distribution of speech features.

The Deep Learning Revolution

- **2010s - Present: Deep Neural Networks (DNNs)** - DNNs, particularly Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), revolutionized speech recognition by automatically learning complex features from vast amounts of data.
- **Emergence of LSTMs, GRUs, and Attention Mechanisms** - Further advancements with specialized architectures like Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), and attention mechanisms significantly improved accuracy and robustness.
- **End-to-End Speech Recognition** - Models directly map acoustic features to text, simplifying the traditional pipeline and achieving state-of-the-art performance.

Deep Learning's impact on speech recognition

- Deep Learning revolutionized speech recognition by addressing key limitations of traditional approaches like GMM-HMM (Gaussian Mixture Model-Hidden Markov Model):

- Feature Learning
  - Traditional: Required hand-crafted features and expert knowledge
  - Deep Learning: Automatically learns relevant features from raw or minimally processed audio

2. Context Understanding
   - Traditional: Limited context window and rigid state transitions
   - Deep Learning: Can capture long-range dependencies and complex patterns in speech

3. Performance Improvements
   - Error rates dropped dramatically (30-50% reduction) when deep neural networks were introduced
   - Particularly better at handling:
     - Noisy environments
     - Different accents and speaking styles
     - Natural, conversational speech

Current applications

1. Virtual Assistants
    - Siri, Alexa, Google Assistant
    - Natural language interfaces for smart home devices
    - In-car voice control systems
2. Business Solutions
    - Automated customer service systems
    - Meeting transcription services
    - Voice biometrics for authentication
3. Healthcare
    - Medical dictation and documentation
    - Remote patient monitoring
    - Voice-based diagnostic tools

Industry trends

1. On-device Processing
   - Shift toward local processing for privacy and reduced latency
   - Lightweight models for mobile devices
2. Multilingual Capabilities
   - Single models handling multiple languages
   - Real-time translation services
3. Personalization
   - Voice recognition systems adapting to individual users
   - Custom wake words and commands
4. Integration with Other Technologies
   - Multimodal systems combining voice with vision
   - Integration with AR/VR environments

Key challenges in speech processing with DNN

1. Signal Variability
    - Speaker characteristics (accent, age, gender)
    - Environmental noise and acoustics
    - Recording conditions and hardware differences
2. Model Engineering
    - Handling variable-length inputs
    - Real-time processing requirements
    - Balancing accuracy vs computational cost
    - Memory constraints for deployment
3. Training and Technical Challenges
    - Large data requirements
    - Class imbalance in training data
    - Handling disfluencies (um, uh, stutters)
4. Domain Adaptation
    - Generalizing across different contexts
    - Handling unseen speakers/conditions
    - Adapting to new languages/accents
    - Transfer learning effectiveness

2. Core Architectures for Speech Recognition

- RNN
- LSTM & GRU
- CNN
- Hybryd CNN-RNN

RNN

1. RNN Basics

    - Process sequential data by maintaining hidden state $ht = f(Wx * xt + Wh * ht-1 + b)$
    - Each timestep considers current input and previous state
    - Share parameters across time steps
    - Natural fit for variable-length speech sequences

3. Limitations with Long Sequences
    a) Vanishing Gradients

    - Gradients become extremely small during backpropagation
    - Early inputs lose influence over time
    - Makes learning long-term dependencies difficult
    - Critical issue for long speech utterances

- b) Exploding Gradients
    - Gradients become extremely large
    - Leads to unstable training
    - Can cause numerical overflow
    - Requires gradient clipping

4. Practical Impact on Speech Processing

    - Poor performance on long utterances
    - Difficulty capturing long-range dependencies
    - Limited context window in practice
    - Memory inefficiency for long sequences

LSTM

1.LSTM Mechanism

- Memory cell: Long-term information storage
- Input gate: Controls new information flow
- Forget gate: Removes irrelevant information
- Output gate: Controls information output
- Cell state: Carries information through time

2.LSTM Advantages for Speech

- Handles variable-length sequences
- Captures long-term dependencies
- Resistant to vanishing gradients
- Better at learning speech timing patterns
- Maintains speaker context over time

GRU

3.GRU Mechanism

- Simpler than LSTM
- Reset gate: Controls past state influence
- Update gate: Combines current/past info
- No separate memory cell
- Fewer parameters than LSTM

4.GRU Advantages

- Faster training than LSTM
- Less memory usage
- Similar performance to LSTM
- Better for shorter sequences
- Easier to train with less data

Applications of LSTM& GRU

- Acoustic modelling
- Language modelling
- Speaker verification
- Emotion recognition
- Speech enhancement

CNN

1. Core Concept
    - Treats spectrograms as 2D images
    - Detects time-frequency patterns
    - Uses convolutions and pooling operations
2. Key Benefits
    - Captures local acoustic features
    - Parallel processing capability
    - Translation invariance
    - Efficient parameter sharing
3. Applications
    - Feature extraction
    - Speech enhancement
    - Speaker/phoneme recognition

Hybrid CNN-RNN Architectures (i)

1. Basic Structure
    - CNN layers: Process spectrogram, extract features
    - RNN layers: Model temporal dependencies
    - CNN outputs feed into RNN inputs
    - Final layers for task-specific predictions
2. Why Combine Them
    - CNN: Local feature extraction, noise robustness
    - RNN: Temporal modeling, sequence handling
    - Together: Better than either alone
    - Balanced efficiency and accuracy

Hybrid CNN-RNN Architectures (ii)

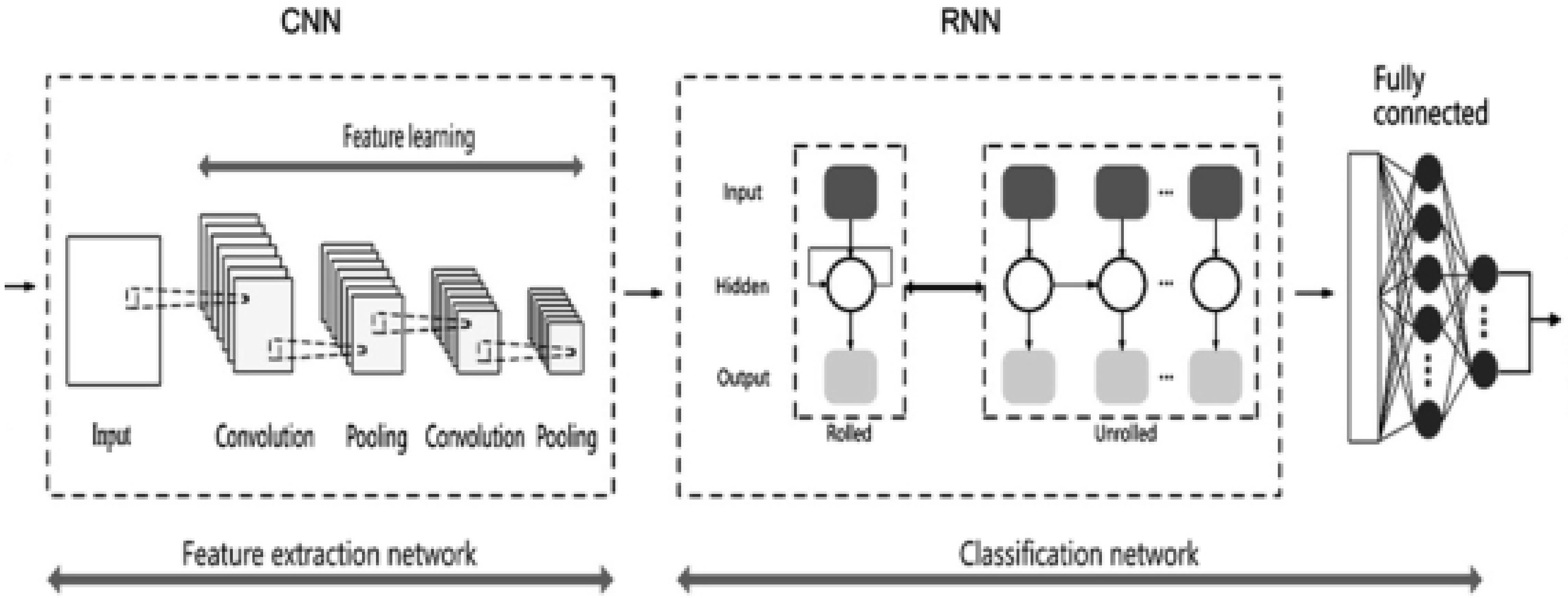
3.Common Variations

- CRNN: Basic CNN followed by RNN
- DeepSpeech2: Multiple CNN+RNN layers
- BiDirectional: RNNs process both directions
- Attention-augmented: Added attention mechanisms

4.Advantages

- Better feature representation
- Reduced computational cost vs pure RNN
- Improved accuracy
- More robust to noise/variations

Hybrid CNN-RNN Architectures (iii)



3. Modern Automatic Speech Recognition Systems

- End-to-End ASR
- Connectionist Temporal Classification (CTC)
- Attention Mechanisms in ASR
- Transformers for Speech Recognition

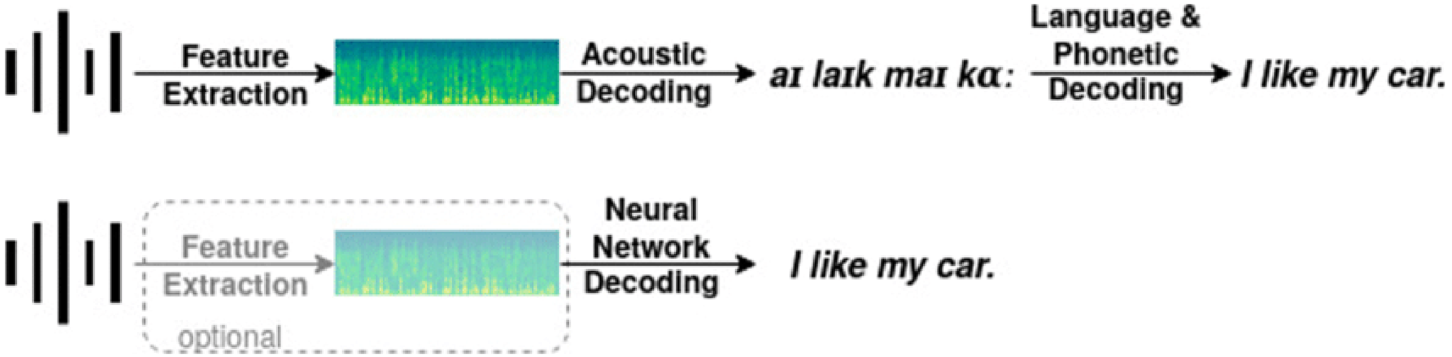
End-to-End ASR Systems vs Traditional Pipeline

1.Traditional Pipeline

- Separate components: acoustic model, pronunciation model, language model
- Each component trained independently
- Complex integration and optimization
- Requires expert knowledge for each component
- Error propagation between stages

2.End-to-End Approach

- Single neural network
- Direct audio-to-text conversion
- Joint optimization of all components
- Trained on <audio, transcript> pairs
- Simpler training and deployment



End-to-End ASR Systems vs Traditional Pipeline

3. Key Advantages
    - Reduced system complexity
    - Better overall optimization
    - Easier to update and maintain
    - Fewer hand-designed components
    - Lower latency potential
4. Main Trade-offs
    - Requires more training data
    - Less interpretability
    - May need larger models
    - Less modular for specific updates

Connectionist Temporal Classification (CTC)

Imagine you need to convert audio to text. The main problem is:

- Audio is long (many frames)
- Words aren't aligned with specific moments
- We don't know exactly which part of audio corresponds to each letter

CTC solves this:

1. Allows the system to "guess" where letters go
2. Uses a special symbol (blank) to:
    1. Separate repeated letters
    2. Fill spaces where there are no letters

Practical example:

- Audio says "hello"
- System might predict: "h-e-ll-o-“ *(where "-" is the blank symbol)*
- It could also be: "-h-eell-oo-"
- All these variations convert to "hello"

Connectionist Temporal Classification (CTC)

- All these paths for “hello” are valid. Each frame in a path has a probability - how confident the system is about that guess.
- The scoring works like this:
  1. For each path, multiply the probabilities of each guess
  2. Add up all the path probabilities that lead to the correct word
  3. Higher total probability means the system is doing a good job
- During training, if the total probability is low, the system adjusts to make better guesses next time. It's like learning to recognize patterns: the more examples it sees, the better it gets at matching sounds to letters.
- Think of it like multiple people trying to write down what they hear. Some might write it slightly differently, but as long as it reads correctly when cleaned up, it's considered correct. The system learns which ways of hearing and writing are most reliable.

Attention Mechanisms in ASR

1. Basic Concept
   - Helps model focus on relevant parts of audio
   - Creates dynamic connections between encoder and decoder
   - Learns where to "pay attention" in input sequence
   - Solves alignment problems better than CTC
2. How It Works
   - Encoder processes audio into features
   - Decoder generates text one symbol at time
   - Attention scores relevant audio parts for each output
   - Weighted sum creates context for prediction
   - Updates focus as it generates each character

Types of Attention

- Bahdanau Attention:
  - Additive attention
  - Uses separate neural network
  - Computes alignment scores with learned weights
  - Better for varying length sequences
  - More flexible but computationally intensive
- Luong Attention:
  - Multiplicative attention
  - Simpler, faster computation
  - Direct dot product between states
  - Works well for similar length sequences
  - More memory efficient

Key Advantages of attention

1. "Better handling of long sequences"
    - Imagine listening to a long sentence. Regular systems might forget the beginning by the time they reach the end
    - Attention is like taking notes - it can look back at any part whenever needed
    - Works better for long speeches or complex sentences
2. "No monotonic alignment assumption"
    - Old systems assumed words must be processed strictly left to right
    - Attention is more like how humans understand speech - we can connect related parts even if they're far apart
    - Example: In "The cat, which I saw yesterday, is black", attention can link "cat" with "is black" even with words in between
3. "Can attend to multiple parts of input"
    - Like having multiple sticky notes while listening
    - Can focus on several important parts at once
    - Example: When hearing "twenty-three", it can look at both parts together to understand it's "23"
4. "Helps with noisy or unclear speech"
    - Like having the ability to "double-check" unclear parts
    - If one part is unclear, can use context from other parts
    - Similar to how we understand someone speaking in a noisy room
5. "Provides interpretable alignment visualization"
    - We can see what parts of speech the system is focusing on
    - Like highlighting the words as they're being processed
    - Helps understand how the system makes decisions

Transformers in Speech Recognition

1. Basic Structure
   - Uses self-attention instead of recurrence
   - Parallel processing of entire sequence
   - Encoder-decoder architecture
   - Positional encoding for timing
2. Key Advantages vs RNNs
   - Faster training (parallel processing)
   - Better at long dependencies
   - No vanishing gradient problems
   - Captures global relationships easier
   - More stable training
3. Trade-offs
   - Needs more memory
   - Requires more training data
   - More complex architecture
   - Higher computational cost

4. Practical Considerations and Future Directions

- Data Augmentation for Speech
- Transfer Learning in Speech Models
- Model Optimization and Deployment
- Future trends

Data Augmentation for Speech

1. Noise Injection
    - Adds different types of background noise
    - Examples: café noise, street sounds, music
    - Helps model handle real-world conditions
    - Controls noise level (Signal-to-Noise Ratio)
    - Makes system more robust
2. Speed Perturbation
    - Changes speech rate (faster/slower)
    - Usually 0.9x to 1.1x speed
    - Maintains pitch and intelligibility
    - Creates more training examples
    - Helps handle different speaking rates
3. Time Stretching
    - Changes duration without changing pitch
    - Like slow-motion or speed-up
    - Preserves speech characteristics
    - Different from speed perturbation
    - Good for accent/dialect variations

Data Augmentation for Speech

4.Other Common Techniques

- Pitch shifting
- Volume adjustment
- Room simulation (reverb)
- Frequency masking
- Time masking

5.Benefits

- Increases dataset size
- Improves model robustness
- Reduces overfitting
- Better real-world performance
- Handles varied conditions

Transfer Learning in Speech Models

1. Basic Concept
   - Start with model trained on large dataset
   - Reuse learned features for new task
   - Adapt to specific needs with less data
   - Like teaching someone who already knows basics
2. How It Works
   - Pre-training phase:
     - Train on huge general dataset
     - Learn basic speech patterns
     - Develop feature understanding
   - Fine-tuning phase:
     - Adjust for specific task
     - Use smaller target dataset
     - Keep useful knowledge

Transfer Learning in Speech Models

3. Common Approaches
    - Feature extraction only
    - Partial model fine-tuning
    - Full model fine-tuning
    - Layer-by-layer adaptation
4. Benefits
    - Needs less training data
    - Faster training time
    - Better performance
    - Works for low-resource languages
    - Cost-effective
5. Applications
    - Speech recognition
    - Speaker identification
    - Emotion detection
    - Accent adaptation
    - Language transfer

Model Optimization and Deployment

1. Quantization
   - Reduces numerical precision
   - Examples:
     - Float32 to Float16/Int8
     - Fewer bits per weight
   - Benefits:
     - Smaller model size
     - Faster inference
     - Less memory usage
   - Trade-off: Slight accuracy loss
2. Pruning
   - Removes unnecessary connections
   - Methods:
     - Weight pruning
     - Channel pruning
     - Structured pruning
   - Process:
     - Identify less important weights
     - Remove them
     - Retrain network

Model Optimization and Deployment

3. Knowledge Distillation
   - Creates smaller "student" model
   - Learns from larger "teacher" model
   - Maintains most performance
   - Reduces model complexity
4. Model Compression
   - Weight sharing
   - Huffman coding
   - Matrix factorization
   - Low-rank approximation
5. Deployment Optimization
   - Hardware-specific optimization
   - Batch processing
   - Caching strategies
   - Pipeline optimization
   - Real-time considerations

Future trends

1. Multimodal Learning
    - Combines speech with:
        - Video (lip reading)
        - Text
        - Gestures
    - Improves accuracy in noisy environments
    - Better context understanding
    - More natural interaction
2. Unsupervised Learning
    - Learning from unlabeled data
    - Self-supervised pre-training
    - Reduces dependency on labeled data
    - Better representation learning
    - Cross-lingual transfer

Future trends

3. Low-Resource Scenarios
    - Solutions for rare languages
    - Few-shot learning
    - Cross-lingual transfer
    - Data augmentation techniques
    - Active learning approaches
4. Emerging Directions
    - End-to-end multilingual models
    - Real-time translation
    - Emotion understanding
    - Personalization
    - Privacy-preserving ASR
5. Technical Advances
    - Smaller, efficient models
    - On-device processing
    - Continuous learning
    - Zero-shot capabilities
    - Better noise handling

LINKS

Speech Emotion Recognition using LSTM and RNN

- Code
- https://github.com/Utkarsh2812/Speech-Emotion-Recognition-Using-LSTM-and-RNN/blob/main/Speech_emotion_recognition_lstm%26rnn.ipynb

- Toronto emotional speech set
- https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess

Tensorflow Speech Recognition Demo

- https://github.com/llSourcell/tensorflow_speech_recognition_demo

- Data is not available