**Speech Synthesis, LPCNet**

- Use of Computers to synthesize intelligible speech
- DSP (Linear Prediction), Deep Learning (WaveRNN) \(\Rightarrow\) LPCNet

**Features:** Bark Cepstrum

Frame Rate Network, Sample Rate Network
- Signal \((s,t)\), Excitation \((e,t)\), Prediction \((p,t)\)
- Output: Excitation \((e,t)\)

**Minimize cross-entropy loss**

---

**Why move ahead?**

**Explicit need of LPCs**
- Need clean acoustic features which might not always be available

**Why not “Learn” the LPCs?!**
- Learn LPCs directly from the input features
  - Possible better fit
- Not restricted to clean speech features
  - Arbitrary codec features (or more general neural features)
- Opens up LPCNet to end-to-end tasks like TTS or speech coding

---

**End-to-end LPCNet**

1. **Learning the LPCs**
   - Using subset of FRN features to avoid overhead
   - Learn Reflection Coefficients instead of LPC for stability
   - Levinson recursion to convert RC \(\Rightarrow\) LPC

2. **Differentiable Embedding Lookup**
   - \(\mu\)-law quantized inputs prevent gradient backpropagation
   - Linearly interpolate between adjacent embeddings
   
   \[
   v^{(i)}(x) = (1 - f) \cdot v_{[x]} + f \cdot v_{[x]+1} 
   \]

   \[
   f = x - [x] 
   \]

3. **Modified Loss Function**
   - LPCNet uses precomputed excitations,
     - Have to compute real-valued excitations on the fly
   - Interpolate excitation probabilities linearly (to propagate gradients)
   - Naïve cross entropy minimization \(\Rightarrow\) network cheats by forcing excitation to be large (\(\mu\)-law spacing wider for larger excitation)
   - To overcome this, compute cross-entropy in linear domain

   \[
   L = -\left(p_{t} \log p_{t} + (1 - p_{t}) \log (1 - p_{t})\right) + \alpha \log \left(\frac{256}{128}\right) \left(p_{t} - s_{t}\right) 
   \]

**Need for Regularization**

- Prevent divergence + improve performance
  - \(L_{1}\): Increase the weight on the second term in the loss
  - Log-Area Ratio (LAR): Match network predicted LPCs to ground truth
    - Network predicts Reflection Coefficients (RCs)
    - Transform to Log-Area ratio (LAR) and match with ground truth LAR
  - Log-Area Ratio matching with naïve cross-entropy minimization (LAR/CE)
    - Minimize \(\mu\)-law cross-entropy to prevent divergence

---

**Final Model**

- End-to-end LPCNet that can learn the LPCs!
  - Not restricted to inputs from which we have to obtain the LPCs
  - Improved Loss + Regularization for better performance
  - Use LPCNet for broader range of applications
    - Speech Enhancement, TTS etc.

**References**