

Abstract

- ▶ PyHTK is a Python-based library and associated pipeline to construct large-scale complex ASR systems using HTK.
- ▶ PyHTK can generate ANNs with versatile model architectures by converting a configuration file into an HTK model.
- ▶ PyHTK and its ASR pipeline integrates a range of capabilities for ASR training and testing based on HTK 3.5.1.
- ▶ Python-based ASR pipeline is suitable for different data sets and allows for both step-by-step and automatic end-to-end use.

Hidden Markov Model Toolkit (HTK)

- ▶ HTK was first developed in 1989 at CUED.
- ▶ HTK has integrated many HMM-based techniques such as
 - ▶ Phonetic decision trees for HMM state-tying;
 - ▶ Transform-based speaker adaptation;
 - ▶ Lattice-based discriminative sequence training.
- ▶ HTK 3.5 released in 2015 includes native support for ANN AMs.
 - ▶ ANN structure can be any directed cyclic graph;
 - ▶ ANN parameters can be shared and speaker dependent;
 - ▶ Data cache supports commonly seen data assessments;
 - ▶ Can use all previous HTK functions seamlessly.
- ▶ Tutorials and details of use are documented in HTKBook.

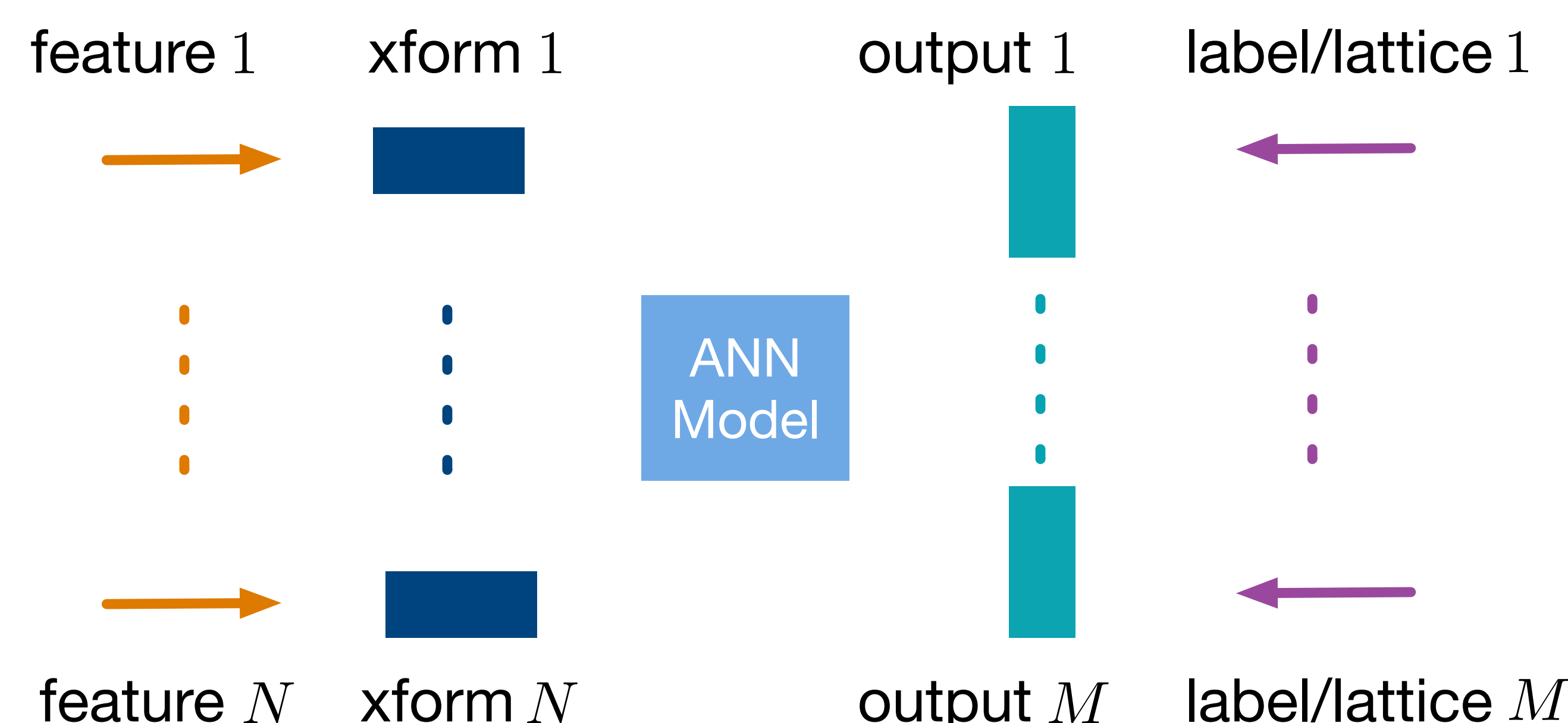
New Features in HTK 3.5.1

- ▶ Feature mixture is extended for more types of operations.
 - ▶ *Concatenation*: for TDNNs & RNNs etc.
 - ▶ *Addition*: for residual connections etc.
 - ▶ *Multiplication*: for gating mechanism for LSTM etc.
 - ▶ *Scaling*: for attention mechanism etc.
- ▶ Example ANN layers supported by HTK 3.5.1 include:

Layer Type	Description
FC	Conventional fully connected layer
LSTM	LSTM model for one time step
GRU	GRU model for one time step
CNN	2D convolution with rectangle shaped filters
GMM	Every output unit is a separate GMM
SelfAttentive	Self attention with modified penalty terms

- ▶ Many new training/test facilities added, e.g. low-frame-rate HMM, multi-channel raw waveform, and ML sequence training etc.

- ▶ Multiple sets of input features and output layers with different labels/lattices are allowed.



PyHTK Library

- ▶ PyHTK can create any HTK model structure using a config file.

```
[ModelSet]
@FeatureType = <FBANK_D_Z>
InputObservation.Type = @FeatureType
@FeatureDim = 80
InputObservation.Dim = @FeatureDim
@RecurrentDim = 500
[NVector:ZeroVec]
Length = @RecurrentDim
Values = 0.0
[Layer:layer_rnn]
Kind = RNN
FeatureMixture.Num = 2
FeatureElement1.Dim = @FeatureDim
FeatureElement1.ContextShiftSet = {+5}
FeatureElement1.Source = @FeatureType
FeatureElement2.Dim = @RecurrentDim
FeatureElement2.ContextShiftSet = {0}
FeatureElement2.Source = ~V ZeroVec
UnfoldValue = 20
OutputDim = @RecurrentDim
ActivationFunction = GatedSigmoid
[Layer:layer_out]
Kind = FC
FeatureMixture.Num = 1
FeatureElement1.Dim = @RecurrentDim
FeatureElement1.ContextShiftSet = {0}
FeatureElement1.Source = layer_rnn
OutputDim = @auto
ActivationFunction = Softmax
[NeuralNetwork:RNN1L]
Layer2.Name = layer_rnn
Layer3.Name = layer_out
```

- ▶ Instead of an HTK wrapper in Python, PyHTK is standalone by re-implementing some HTK C library functions in Python.
 - ▶ Interacts with HTK by reading/writing HTK data/model.
 - ▶ Integrated with NumPy and can use other Python-NumPy interfaces (e.g. PyTorch or TensorFlow) easily.
- ▶ Is maintained via GitHub and documented by Sphinx Autodoc.

PyHTK Pipeline

- ▶ PyHTK ASR pipeline is adjustable according to different config files and can be applied to different speech data sets.
- ▶ Example PyHTK pipeline training & test steps include:

Step	Description
xform	Linearly transform GMM-HMMs
sat	Speaker adaptive training with CMLLR
mpe	MPE/MMI training for GMM-HMMs
dnn-ce	CE training for ANN-HMMs
dnn-mpe	MPE/MMI training for ANN-HMMs
align	Generate alignments using Viterbi algorithm
decode	Decode a set of HMM acoustic models
rescore-am	Rescore lattices with different acoustic models
rescore-lm	Rescore lattices with a different setup or LM

275 Hour MGB3 Experiments

- ▶ Viterbi decoding results of some selected HTK ANN models with a 3-gram LM and a 63k word vocabulary on dev17b are listed.

System	Criterion	%WER
DNN	CE	28.4
LSTMP	CE	25.7
HORNNP	CE	25.6
ResNet-TDNN	CE	25.1
BD-FD-GridRNN-ResNet-TDNN	CE	24.6
DNN-GMM	MPE	26.0
DNN-GMM+CMLLR SAT	MPE	25.1
BD-FD-GridRNN-ResNet-TDNN	MPE	22.7

Summary

- ▶ PyHTK includes our recently developed Python library and the corresponding Python based ASR pipeline.
- ▶ More functions will be included in future HTK & PyHTK releases.