Analysis of deeplearning for a limited vocabulary - Phase 2 Model Selection

Summary **– This report depicts outcomes of a study on model selection by utilizing various approaches. Neural Networks and Hidden Markov Model were used as model trainers.**

1 Introduction

Given a data-set of 41 context based words, 30620 entries, we're supposed to build a final system to classify words in real-time in flight simulator. For doing so, we built 3 different models, namely; NNs, 1-vs-all NNs, HMMs.

1. NN
   Multi-layer Neural Network: One model for whole data-set/class.
   Input dense: 41 categorical : [0,....,1,0], output dense: 41 categorical [0,....,1,0]
   Used API: Keras
2. 1-vs-all NNs
   Multi-layer Neural Network: Each 41 class trained by applying one-vs-all on a data-set.
   Input dense: binary categorical : [0,1], output dense: binary categorical [0,1]
   Used API: Keras
3. HMMs
   Sequential Markov Model: Each 41 class trained with its data.
   Input: n$\times$13 (n is sequence length), output: logprob
   Used API: hhmlearn

2 Dataset

Out data-set contains 30620 entries of 41 context based words. Each entries saves mffc features of given audio file.

Table 1: Data-set after split

Train	Test
26027	4593

* models were trained by enabling cross validation in each epoch, thus we did not split our data into extra validation set.

3 NN Model

Following code shows our model structure. Model have 3 layers. Dense of input and hidden layer is 200. Softmax activation was used at the output layer as our model is categorical.

Input and output dense for the model:
Input dense: 41 categorical : [0,....,1,0], output dense: 41 categorical [0,....,1,0]

```
model = Sequential()
#add layers to model
model.add(Dense(200, activation='sigmoid', input_shape=(n_cols,)))
model.add(Dense(200, activation='sigmoid'))
model.add(Dense(41, activation='softmax'))
```

Listing 1: Initialize model

3.1 Parameters

Table 2: Optimized parameters for model

Parameters	Values Searched	Best
activation	relu, sigmoid	sigmoid
optimizer	adam, SGD	adam
loss		categorical_crossentropy
epochs		50
batch_size		20
validation split		0.2

3.2 Results

For more accurate evaluation of model we conducted extra layer of evaluation by adding different metrics such as setting threshold values for the acceptance recognition. That approach enabled us to see the rejection values of each class.

1. Accuracy on unseen test data (mix model)
   - Normal Accuracy: 98.62%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$): 97.69%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$ and $\overline{y}_k - \widetilde{y}_p \geq \Delta_2$): 97.69%

Aforementioned intuitions can be described as follows:

$\overline{y}_k$ and $\widetilde{y}_p$ are the maximum two elements of the output vector.

$$\overline{y}_k = \max_{1 \leq i \leq N} y_i, \quad k = \arg\max_{1 \leq i \leq N} y_i. \tag{1}$$

$$\widetilde{y}_p = \max_{i \leq i \leq k; k+1 \leq i \leq N} y_i \tag{2}$$

where $\Delta_1 = 0.9$ and $\Delta_2 = 0.5$

If any of the threshold conditions did not meet on a found class, then it labeled as rejected sample.

1. Accuracy with rejection
   - Accuracy with rejection: 97.63%
   - Total rejection (false true): 1.66%

- Error rate (true false): 0.71%

Table 3: Rejection values for each class

target	Rej value	target	Rej value	target	Rej value
zero	0.0000	two	0.0000	eight	0.0000
cabin doors	0.0000	decimal	0.0000	tow-bar	0.0000
adjusted & locked	0.0000	five	0.7353	six	0.8065
four	0.8403	all switches	0.8475	fuel selector	0.8547
cockpit checklist completed	0.8772	abroad	0.9091	three	0.9524
shut-off cabin heat	1.0309	closed	1.1628	nine	1.6393
on	1.1667	removed	1.6667	weight and balance	1.7699
one	1.757	fuel quantity	1.8018	flight controls	1.1849
alternate air door	1.8519	seats & belts	2.0619	preflight inspection	2.0833
circuit breakers	2.5424	off	2.3622	open	2.4194
fuel shutoff valve	3.1250	fuel temperature	3.1579	battery main busy	3.1915
locked	3.2787	completed	3.3058	seven	3.3333
cockpit	4.0650	in	4.6269	checked	0.0000
ac documents	2.5424	sufficient	2.9412		

4 1-vs-all NNs Model

Following code shows our model structure. In total, model have 4 layers. Dense of input and hidden layer is 200. Softmax activation was used at the output layer as our model is categorical.

Input and output dense of models:
Input dense: binary categorical : [0,1], output dense: binary categorical [0,1]

```
# define model
model = Sequential()
#add layers to model
model.add(Dense(200, activation='relu', input_shape=(n_cols,)))
model.add(Dense(200, activation='relu', input_shape=(n_cols,)))
model.add(Dense(200, activation='relu'))
model.add(Dense(2, activation='softmax'))
```

Listing 2: Initialize 1-vs-all model

4.1 Parameters

Table 4: Optimized parameters for model

Parameters	Values Searched	Best
activation	relu, sigmoid	relu
optimizer	adam, SGD	adam
loss		categorical_crossentropy
epochs		20
batch_size		20
validation split		0.2

4.2 Results

When it comes to evaluating model, given each test data, we run all the models on a new data and pick the one with the best score. The whole process can be descried as follows:

$y = \{y_0, y_1, ...y_N\}$ is the set of the output probability of all single Multilayer Neural Networks on single entity.

$$k = \arg\max_{1 \leq i \leq N} y_i \tag{3}$$

For fair evaluation of model generalization we conducted extra layer of evaluation by adding different metrics such as setting threshold values for the acceptance recognition. That approach enabled us to see the rejection values of each class.

1. Accuracy on unseen test data (mix model)
   - Normal Accuracy: 98.62%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$): 97.69%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$ and $\overline{y}_k - \widetilde{y}_p \geq \Delta_2$): 97.69%

If any of the threshold conditions do not meet on a found class, then it labeled as rejected sample.

1. Accuracy with rejection
   - Accuracy with rejection: 97.00%
   - Total rejection (false true): 2.8667%
   - Error rate (true false): 0.13%

Table 5: Rejection values for each class

target	Rej value	target	Rej value	target	Rej value
three	0.0000	circuit breakers	0.0000	abroad	0.0000
five	0.7353	all switches	0.8475	fuel quantity	0.9009
cabin doors	0.9434	decimal	1.5504	nine	1.6393
four	1.6807	all fuel selector	1.7094	zero	1.8018
two	1.8018	checked	1.8018	seat & belts	2.0619
fuel shutoff value	2.0833	battery main bus	2.1277	eight	2.1898
off	2.3622	seven	2.5000	ac documents	2.5254
tow-bar	2.5000	wight and balance	2.6549	in	2.7778
sufficient	2.9412	shut-off cabin heat	3.0928	six	3.2258
open	3.2258	closed	3.4884	one	3.5714
adjusted & locked	3.7383	completed	4.1322	removed	4.1667
preflight inspection	4.5455	alternate air door	4.6296	cockpit checklist completed	5.2632
fuel temperature	5.2632	flight controls	5.5046	cockpit	5.6911
on	5.8333	locked	6.5575		

5 HMM Model

Our data set contains 41 different words, where each word has many audio files associated with it. We have built an HMM model for each class by training our model on given dataset. Then after build model, given new input file, we need to run all the models on this file and pick the one with the best score.

5.1 Parameters

Table 6: Parameters of model

Parameters	Values
algorithm	vitebri
n_iter	1000
covariance	diag cov matrix
params	stmc*

stmc* = Controls which parameters are updated in the training process. Can contain any combination of s for startprob, t for transmat, m for means and c for covariance. Defaults to all parameters.

5.2 Result

$y = \{y_0, y_1, ...y_N\}$ is the set of the *logprob** of all HMM models on single entity.

$$k = \arg\max_{1 \le i \le N} y_i \tag{4}$$

*logprob**: The log probability of the data.

1. Result based on *logprob*
   - Accuracy on train data: 96.12%
   - Accuracy on train data: 95.12%

6 Further Evaluation

Futures evaluations have been done on different data-sets to have more insight regarding the further evaluations and understating biases. Evaluation was conducted on pretrained NN models.

```
DATA-SETS
├── AMMA: Aviation Academy data - n=21433
├── ADA: ADA University data - n=9187
│   ├── ADA Girls - n=7218
│   └── ADA Boys - n=1969
└── MIX: All together - n=30620
```

6.1 AMMA Model

Model evaluation of AMMA (model: Neural Network) on AMMA test and ADA datasets.

1. Accuracy on unseen AMMA Test
   - Normal Accuracy: 98.94%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$): 98.51%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$ and $\overline{y}_k - \widetilde{y}_p \geq \Delta_2$): 98.51%
2. Accuracy on unseen ADA
   - Normal Accuracy: 84.04%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$): 78.29%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$ and $\overline{y}_k - \widetilde{y}_p \geq \Delta_2$): 78.21%

6.2 ADA Girls Model

Model evaluation of ADA Girls (model: Neural Network) on ADA Girls test and ADA Boys.

1. Accuracy on unseen ADA Girls Test
   - Normal Accuracy: 97.28%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$): 94.85%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$ and $\overline{y}_k - \widetilde{y}_p \geq \Delta_2$): 94.85%
2. Accuracy on unseen ADA Boys
   - Normal Accuracy: 76.07%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$): 66.82%
   - Accuracy where ($\overline{y}_k \geq \Delta_1$ and $\overline{y}_k - \widetilde{y}_p \geq \Delta_2$): 66.72%