HOW TO TRAIN A DISCRIMINATIVE FRONT END WITH STOCHASTIC GRADIENT DESCENT AND MAXIMUM MUTUAL INFORMATION

Jasha Droppo, Milind Mahajan, Asela Gunawardana, and Alex Acero

Speech Technology Group
Microsoft Research, Redmond, WA, USA
{jdroppo,milindm,aselag}@microsoft.com

ABSTRACT

This paper presents a general discriminative training method for the front end of an automatic speech recognition system. The SPLICE parameters of the front end are trained using stochastic gradient descent (SGD) of a maximum mutual information (MMI) objective function. SPLICE is chosen for its ability to approximate both linear and non-linear transformations of the feature space. SGD is chosen for its simplicity of implementation. Results are presented on both the Aurora 2 small vocabulary task and the WSJ Nov-92 medium vocabulary task. It is shown that the discriminative front end is able to consistently increase system accuracy across different front end configurations and tasks.

1. INTRODUCTION

The acoustic processing of standard automatic speech recognition systems can be roughly divided into two parts: a front end which extracts the acoustic features, and a back end acoustic model which scores transcription hypotheses for sequences of those acoustic features.

Two outstanding problems with traditional front end design are that the parameters are manually chosen, and that the feature extraction is uniform over the acoustic space.

For the majority of existing systems, the front end parameters are manually chosen based on heuristics such as mel-scale filtering, cepstral liftering, and dynamic feature computation. Choice of parameters is usually a combination of trial and error, and reliance on historical values.

These existing systems are also uniform, in the sense that they extract the same features regardless of absolute location in the acoustic space. This is not a desirable quality, as the information needed to discriminate `/f/` from `/s/` is quite different from that needed to tell the difference between `/aa/` and `/ae/`. A front end with uniform feature extraction must make compromises to get good coverage over the entire range of speech sounds.

This paper demonstrates how stochastic gradient descent (SGD)[1] can be used to discriminatively train the front end parameters, and begin to overcome both of these outstanding problems. The system described is similar to, and a generalization of, the technique for MCE training of stochastic vector mappings in [2].

Until now, discriminative training of the front-end parameters for automatic speech recognition was difficult. Specialized training algorithms needed to be developed for each new parameterization[3, 4, 5]. SGD has the advantage that only the gradient of the objective function with respect to the free parameters needs to be computed, making it easy to implement.

The general idea of jointly training the feature extractor and classifier with stochastic gradient descent is not entirely new. It has, for instance, been shown to improve character recognition[6]. The idea has also been applied to hybrid ANN/HMM speech recognition systems[7]. And, it has also been shown to outperform more complicated batch algorithms[8].

In this paper, the front-end parameters pass through a SPLICE (stereo piecewise linear compensation for environment) [9] transform. Given enough parameters, SPLICE can approximate any feature transformation to an arbitrary precision. As a result, the system presented here is a generalization of any discriminative feature space transformation.

This paper is organized as follows. Section 2 demonstrates how stochastic gradient descent can be applied to train the front end parameters. Section 3 details the procedure for the special case of SPLICE parameters. Experiments on the Aurora 2 and Wall Street Journal tasks are presented in Section 4.

2. TRAINING THE FRONT END WITH SGD

This section presents a general framework for using stochastic gradient descent (SGD) to train the front-end parameters with respect to a maximum mutual information (MMI) objective function.

2.1. MMI Objective Function

For each training utterance $\mathcal{Y}_r$ and corresponding correct transcription w_r, the MMI objective function is the composition of two functions: the feature transformation and the per-utterance objective function.

$$\mathcal{X}_r = f_\lambda(\mathcal{Y}_r) \tag{1}$$

The feature transformation, Eq. 1, provides the transformed input $\mathcal{X}_r$ to the speech recognition system for utterance r from the raw input $\mathcal{Y}_r$.

$$\mathcal{F}_r = \ln \frac{p(\mathcal{X}_r, w_r)}{\sum_w p(\mathcal{X}_r, w)} \tag{2}$$

The per-utterance objective function, Eq. 2, is the log of the conditional probability of the correct transcription under the current acoustic model, given the acoustics. Ideally, the sum in the denominator of Eq. 2 is taken over all permissible transcriptions for the current utterance.

2.2. Computing the Gradient

To use SGD, it is necessary to compute the partial derivative of $\mathcal{F}_r$ with respect to the free model parameters. Fortunately, the structure of the objective function allows a simple application of the chain rule.

Every $\mathcal{F}_r$ is a function of many acoustic model state conditional probabilities $p(x_t^r|s_t^r)$.[1] Each of these is, in turn, a function of the front end transformed features x_{it}^r. And, each transformed feature is a function of the front end parameters λ.

$$\frac{\partial \mathcal{F}_r}{\partial \lambda} = \sum_{t,s,i} \frac{\partial \mathcal{F}_r}{\partial \ln p(x_t^r|s_t^r = s)} \frac{\partial \ln p(x_t^r|s_t^r = s)}{\partial x_{it}^r} \frac{\partial x_{it}^r}{\partial \lambda} \tag{3}$$

Here, r is an index into the training data. The tth observation vector in utterance r is identified by x_t^r. The scalar x_{it}^r is the ith dimension of that vector. The back end acoustic model state at time t in utterance r is s_t^r.

The first term in Eq. 3 captures the sensitivity of the objective function to individual acoustic likelihoods in the model. It can be shown to be equal to the difference of the conditional and unconditional posterior, with respect to the correct transcription. These are simply the flattened numerator and denominator terms that occur in standard lattice-based MMI estimation[10].

$$\begin{aligned} \frac{\partial \mathcal{F}_r}{\partial \ln p(x_t^r|s_t^r = s)} &= p(s_t^r = s|\mathcal{X}_r, w_r) - p(s_t^r = s|\mathcal{X}_r) \\ &= \gamma_{rts}^{\text{num}} - \gamma_{rts}^{\text{den}} \end{aligned} \tag{4}$$

[1] This derivation assumes one Gaussian mixture component per state of the acoustic model. For the multiple mixture component case, the variable s indexes not state, but the individual mixture components. Nothing else needs to be changed.

The second term in Eq. 3 captures the sensitivity of individual likelihoods in the acoustic model with respect to the front end transformed features. Computing this differential is a simple matter.

$$\frac{\partial \ln p(x_t^r|s_t^r = s)}{\partial x_t^r} = -\Sigma_s^{-1}(x_t^r - \mu_s) \tag{5}$$

Here, μ_s and Σ_s are mean and variance parameters from the Gaussian component associated with state s in the back end acoustic model.

The final term in Eq. 3 captures the relationship between the transformed features and the parameters of the front end. Its form heavily influences the final gradient computation, and is derived for SPLICE in Section 3.1.

2.3. Update

The update rule for stochastic gradient descent is quite simple. After shuffling the training data into a random order, utterances are presented one at a time to the training algorithm. As the nth training example is presented, the parameters are updated with a scaled version of the gradient of the objective function with respect to the front end parameters.

$$\lambda_{n+1} = \lambda_n + \eta_n \left. \frac{\partial \mathcal{F}_n}{\partial \lambda} \right|_{\lambda_n}$$

For testing, a smoothed version of the parameters is used. Smoothing alleviates the problem of selecting good values for the η_n [1].

$$\lambda_{\text{test}} = \frac{1}{N} \sum_n \lambda_n$$

For this paper, we chose to use a constant η independent of n. Additionally, mean and variance normalization is applied to the data, so that a single scalar η can be applied to all of the parameters.

Since there is no standard evaluation set defined for the Aurora 2 task, a good value for η was selected by monitoring training set accuracy. Several values of η were evaluated, and chose the value that lead to saturating the training set accuracy after two full passes over the data.

2.4. Approximating the Gradient with Lattices

Eq. 4 requires computation of acoustic model state and mixture component posterior probabilities. Since exact computation can be resource intensive, the posteriors were approximated on word lattices generated by the baseline maximum likelihood acoustic model. The time marks in the lattices were held fixed, and forward-backward was used within each arc to determine arc conditional posterior probabilities.

As is commonly done in lattice-based MMI estimation, the model was also modified to include posterior flattening[10].

3. APPLICATION TO THE SPLICE TRANSFORM

The SPLICE transform was introduced as a method for overcoming noisy speech [9]. It models the relationship between feature vectors y and x as a constrained Gaussian mixture model (GMM).

In this paper, y is a traditional feature vector based on static cepstra and its derivatives. But, it should be possible to expand y to include more context information, finer frequency detail, or other non-traditional features.

We place no constraint on x, other than it is a feature space that improves our objective function. This gives the system more freedom than existing methods that define x as clean speech[9] or phone posteriors[11].

For our purposes, the relationship between x and y can be expressed as a joint Gaussian mixture model (GMM). One way of parameterizing this model is as a GMM on y, and a conditional expectation of x given y and the model state m.

$$\begin{aligned} p(y,m) &= N(y;\mu_m,\sigma_m)\pi_m \\ E[x|y,m] &= A_m y + b_m \end{aligned}$$

The parameters λ of this transformation are a combination of the means μ_m, variances σ_m, and state priors π_m of the GMM $p(y,m)$, and the rotation A_m and offset b_m of the affine transformation.

The SPLICE transform $f_\lambda(y)$ is defined as the minimum mean squared estimate of x, given y and the model parameters λ. In effect, the GMM induces a piecewise linear mapping from y to x.

$$\hat{x} = f_\lambda(y) = E[x|y] = \sum_m (A_m y + b_m) p(m|y) \qquad (6)$$

3.1. SPLICE Gradients

For the case of the full SPLICE transform, the final partial derivative in Eq. 3 with respect to the linear transform parameters A_m can be derived as follows. For element a_{uv} in the matrix A_m, the partial derivative is

$$\begin{aligned} \frac{\partial x_{it}^r}{\partial a_{uvm}} &= \frac{\partial}{\partial a_{uvm}} \sum_{m'} \left(b_{im'} + \sum_j a_{ijm'} y_{jt}^r \right) p(m'|y_t^r) \\ &= 1(i=u) y_{vt}^r p(m|y_t^r) \end{aligned} \qquad (7)$$

Here, $1(z)$ is an indicator function that evaluates to one when z is true, and to zero otherwise.

Combining Eqs. 3, 4, 5 and 7, the gradient of the objective function $\mathcal{F}$ with respect to the matrix A_m is

$$\frac{\partial \mathcal{F}_r}{\partial A_m} = \sum_{t,s} p(m|y_t^r) \left(\gamma_{rts}^{\text{num}} - \gamma_{rts}^{\text{den}} \right) \Sigma_s^{-1} (\mu_s - x_t^r)(y_t^r)^T \qquad (8)$$

The final partial derivative in Eq. 3 with respect to the offset parameters b_m can be derived as follows. For the uth element of the vector b_m,

$$\begin{aligned} \frac{\partial x_{it}^r}{\partial b_{um}} &= \frac{\partial}{\partial b_{um}} \sum_{m'} \left(b_{im'} + \sum_j a_{ijm'} y_{jt}^r \right) p(m'|y_t^r) \\ &= 1(i=u) p(m|y_t^r) \end{aligned} \qquad (9)$$

Combining Eqs. 3, 4, 5 and 7, the complete gradient with respect to the vector b_m is

$$\frac{\partial \mathcal{F}_r}{\partial b_m} = \sum_{t,s} p(m|y_t^r) \left(\gamma_{rts}^{\text{num}} - \gamma_{rts}^{\text{den}} \right) \Sigma_s^{-1} (\mu_s - x_t^r) \qquad (10)$$

3.2. SPLICE with only offsets

In most SPLICE implementations, the conditional mapping of y to x given m is a simple offset ($A_m = I$).

$$\hat{x} = f_\lambda^{\text{off}}(y) = y + \sum_m b_m p(m|y) \qquad (11)$$

For this case of an offset-only SPLICE transform, the total number of front-end parameters can be kept quite low. Training can be accomplished using Eq. 10 only.

3.3. SPLICE with one mixture component

With one mixture component, SPLICE reduces to a simple linear transform. Since there is no requirement that the input and output dimensions be the same, a dimensionality reducing operation can be simultaneously performed.

$$\hat{x} = f_\lambda^{\text{lin}}(y) = Ay \qquad (12)$$

For the case of a single mixture component SPLICE transform, Eq. 8 can be simplified as follows.

$$\frac{\partial \mathcal{F}_r}{\partial A} = \sum_{t,s_t^r} \left(\gamma_{s_t^r}^{\text{num}} - \gamma_{s_t^r}^{\text{den}} \right) \Sigma_{s_t^r}^{-1} (\mu_{s_t^r} - x_t^r)(y_t^r)^T \qquad (13)$$

3.4. Initial Values for SPLICE Parameters

Initial values for the rotation and offset parameters of the SPLICE transform are chosen to correspond to an identity transform of the input data. This ensures that, at the start of discriminative training, the front end and back end are well matched.

Values for the parameters of the SPLICE GMM were initialized by selecting M vectors uniformly spaced throughout the training data. The variance parameters were initialized to unit covariance, and tied across all mixture

components. Ten iterations of expectation-maximization training were then performed to refine the model parameters.

Although the framework would easily enable updating the SPLICE GMM parameters at the same time as the rotation and offset parameters, they were held fixed for the experiments presented in this paper.

4. RESULTS

To demonstrate the effectiveness of the discriminative front end technique, we applied it to existing strong maximum likelihood baselines. After the baseline was constructed, the front end parameters were tuned to improve the accuracy of the overall system. Even against these good baseline systems, the benefits of the discriminative front end are quite apparent.

4.1. The Aurora 2 Baseline

The experiments presented here were based on the data, code, and training scripts provided within the Aurora 2 task[12]. The task consists of recognizing strings of English digits embedded in a range of artificial noise conditions.

The acoustic model (AM) used for recognition was trained with the standard "complex back end" Aurora 2 scripts on the multi-condition training data. This data consists of 8440 utterances, and includes all of the noise types seen in test set A, at a subset of the SNR levels.

The AM contains eleven whole word models, plus `sil` and `sp`, and consists of a total of 3628 diagonal Gaussian mixture components, each with 39 dimensions.

Each utterance in the training and testing sets was normalized using whole-utterance Gaussianization. This simple cepstral histogram normalization (CHN) method provides us with a very strong baseline.

This baseline is better than most published numbers on this task, including the ETSI Advanced Front-End which has an accuracy of 93.24%[13]. Consequently, even small gains represent strong experimental results.

4.2. Offset-only SPLICE (Aurora 2)

Table 1 summarizes the results for this experiment. On top of the CHN baseline, three different SPLICE model sizes were evaluated. Test set accuracy is generally higher on Set A, which is unsurprising considering its similarity to the training data.

Both the 256 and 1024 model sizes achieve a 93.66% accuracy. This essentially matches the best published result for the Aurora 2 task (93.69% [14]), which used the discriminative Tandem acoustic features[11] with mean and variance normalization. All of these results are achieved without modifying the back end structure or training regimen.

Components	Set A	Set B	Set C	Average
CHN Baseline	93.62	93.20	93.52	93.43
64	93.84	93.23	93.55	93.54
256	93.92	93.33	93.79	93.66
1024	93.94	93.33	93.75	93.66

Table 1. Offset-only SPLICE accuracy on Aurora 2 for 64, 256, and 1024 SPLICE mixture components after two full epochs of training. Average accuracy is computed across 0 dB to 20 dB for all test sets.

The recognition accuracy tends to increase as the number of SPLICE parameters are increased, but there is little difference between the 256 and the 1024 model sizes. This seems to indicate that adding even more parameters will not help. One possible reason for this is the high accuracy achieved on the training set. With 1024 mixture components, only 183 errors remain out of 27,727 words. The discriminative objective function has almost no data remaining from which to learn.

4.3. The Wall Street Journal Baseline

The experiments presented here are on the November 1992, 20k test set of the Wall Street Journal (WSJ) task.

The SI84 training data set was used to build the back end acoustic model. The acoustic model contains 91,368 Gaussian components in 4,566 states representing 40 phones with cross-word triphone and word boundary contexts.

The mean and variance of the training data was computed, and used to normalize both the testing and training data.

4.4. Offset-only SPLICE (WSJ)

Table 2 was generated by training a small (256 mixture component) offset-only SPLICE front end for the WSJ task. The discriminative front end was able to reduce the word error rate by 0.3% absolute, or just over 3% relative.

These results demonstrates that, even when the number of parameters in the front end is quite small, the discriminative front end can improve the accuracy of a medium sized vocabulary task.

4.5. Linear Dimensionality Reduction

To demonstrate the effectiveness of adding rotation and dimensionality reduction to SPLICE, a simple single mixture component discriminative front end was built. The transformation rotates a 52-dimensional feature (static, plus first,

Epoch	Error Count	Error Rate
0.00	514	9.1
0.25	514	9.1
0.50	511	9.1
0.75	507	9.0
1.00	506	9.0
1.25	502	8.9
1.50	496	8.9
1.75	500	8.9
2.00	498	8.8

Table 2. Offset-only SPLICE performance on WSJ Nov92 test set.

Epoch	Set A	Set B	Set C	Ave.
0.0	93.62	93.20	93.53	93.43
0.2	93.70	93.12	93.50	93.43
0.4	93.76	93.24	93.61	93.52
0.6	93.80	93.29	93.59	93.55
0.8	93.77	93.34	93.62	93.57
1.0	93.81	93.34	93.66	93.59
1.2	93.83	93.35	93.68	93.61
1.4	93.84	93.34	93.64	93.60
1.6	93.83	93.34	93.64	93.59
1.8	93.81	93.33	93.60	93.58
2.0	93.83	93.33	93.62	93.59

Table 3. Linear dimensionality reduction accuracy on Aurora 2. Results are average across 0 dB to 20 dB conditions.

second, and third order regression coefficients) into a 39-dimensional vector. As the baseline system was trained on a 39-dimensional vector consisting of static, first, and second regression coefficients, the initial SPLICE parameters could be chosen to provide a good match.

Two full epochs of MMI-SPLICE were run, in which the front end matrix was tuned to give more discriminable information to the back end, using all 52 dimensions as input. The result is shown in Table 1. Even though there are only 2028 parameters to tune, and the front end is adding information the back end never saw during training (third order regression coefficients), there is improvement across all three test sets.

5. SUMMARY

We have presented a framework for training a discriminative front-end for noise robust speech recognition, and evaluated it on both the Aurora 2 and Wall Street Journal tasks. Within this framework, we have shown how to improve recognition accuracy with:

- Linear dimensionality reduction transforms of an expanded feature space.
- Non-linear warpings of a fixed feature space.

To expand upon this initial result, future work should include:

- Adding smoothness constraints that we know improve ML-SPLICE.
- Training state-conditional or tied rotation matrices in addition to the offsets.
- Updating GMM parameters to improve MMI criterion.
- Joint acoustic model and front end training.

6. ACKNOWLEDGMENTS

The authors would like to thank Patrick Nguyen, who provided the reference lattice-based MMI code that was used as a basis for the MMI-SPLICE training, as well as the ML baseline for the Wall Street Journal task.

7. REFERENCES

[1] Harold J. Kushner and G. George Yin, *Stochastic Approximation Algorithms and Applications*, Springer, 1997.

[2] J. Wu and Q. Huo, "An evinronment compensated minimum classification error training approach and its evaluation on Aurora 2 database," in *Proc. ICSLP 2002*, Denver, USA, September 2002, pp. 453–457.

[3] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Solatu, and G. Zweig, "FMPE: Discriminatively trained features for speech recognition," in *Proc. 2005 ICASSP*, Philadelphia, USA, March 2005, vol. 1, pp. 961–964.

[4] J. Droppo and A. Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," in *Proc. Interspeech 2005*, Lisbon, Portugal, September 2005.

[5] L. Deng, J. Wu, J. Droppo, and A. Acero, "Analysis and comparison of two speech feature extraction/compensation algorithms," *IEEE Signal Processing Letters*, 2005, Accepted for publication.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[7] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden Markov model hybrid," *IEEE Transactions on Neural Networks*, vol. 3, no. 2, pp. 252–259, 1992.

[8] A. Gunawardana, M. Mahajan, and A. Acero, "Hidden conditional random fields for phone classification," in *Proc. Interspeech 2005*, Lisbon, Portugal, September 2005.

[9] J. Droppo, L. Deng, and A. Acero, "Evaluation of SPLICE on the Aurora 2 and 3 tasks," in *Proc. ICSLP 2002*, Denver, USA, September 2002, pp. 29–32.

[10] Daniel Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University, 2003.

[11] D. Ellis and M. Gomez, "Investigations into tandem acoustic modeling for the Aurora task," in *Proceedings of Eurospeech*, 2001, pp. 189–192.

[12] H. G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluations of speech recognition systems under noisy condidions," in *ISCA ITRW ASR2000*, Paris, France, September 2000.

[13] D. Macho, L. Mauuary, B. Noé, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," in *Proc. ICSLP 2002*, Denver, USA, September 2002, pp. 17–20.

[14] C.-P. Chen and Jeff Bilmes, "Speech feature smoothing for robust ASR," in *ICASSP 2005*, Montreal, 2005, vol. 1, pp. 525–528.