ANNEALED DROPOUT TRAINED MAXOUT NETWORKS FOR IMPROVED LVCSR

Steven J. Rennie, Pierre L. Dognin, Xiaodong Cui, and Vaibhava Goel

IBM Thomas J. Watson Research Center
NY, USA
{sjrennie, pdognin, cuix, vgoel}@us.ibm.com

ABSTRACT

A significant barrier to progress in automatic speech recognition (ASR) capability is the empirical reality that techniques rarely ”scale”—the yield of many apparently fruitful techniques rapidly diminishes to zero as the training criterion or decoder is strengthened, or the size of the training set is increased. Recently we showed that annealed dropout—a regularization procedure which gradually reduces the percentage of neurons that are randomly zeroed out during DNN training—leads to substantial word error rate reductions in the case of small to moderate training data amounts, and acoustic models trained based on the cross-entropy (CE) criterion [1]. In this paper we show that deep Maxout networks trained using annealed dropout can substantially improve the quality of commercial-grade LVCSR systems even when the acoustic model is trained with sequence-level training criterion, and on large amounts of data.

Index Terms— Maxout Networks, Deep Neural Networks, Deterministic Annealing, Dropout Training, Model aggregation.

1. INTRODUCTION

Recently it has been shown that when training neural networks on a limited amount of data, randomly zeroing, or ”dropping out” a fixed percentage of the outputs of a given layer for each training case can improve test set performance significantly [2] . Dropout training prevents the detectors in the network from co-adapting, and so encourages the discovery of approximately independent detectors, which in turn limits the capacity of the network and prevents overfitting. The technique, moreover, represents an extreme form of model aggregation. For log linear models, the geometric average over all 2^N possible models (dropout masks) that can be formed from N feature inputs can be computed by simply re-scaling the outputs. This aggregate model is used during testing, and is guaranteed to have lower cross-entropy than the average cross-entropy of the composite models [3]. More generally for deep networks, no such results have been proven, but the technique works well in practice, particularly for deep neural networks that are conditionally linear, such as rectified-linear (ReLU) [4] and Maxout networks [5] .

Conventional dropout improves test-time performance when there is limited data relative to the size of the model being trained. Dropout training allows one to gain performance by avoiding overfitting, so that a larger model than would otherwise maximize test performance can be utilized. But what about the more usual scenario where the size of the model and training time, rather than the amount of training data, are the dominant constraints? This question has not been previously studied.

Recently in [1] we showed that *annealing* the dropout rate over the course of training can substantially improve the quality of the resulting acoustic model in the case of low to moderate amounts of training data (10-100 hrs). Specifically, we showed that our best Maxout and Sortout [6] networks, which are trained using annealed dropout, outperform the best published WER results on the Aurora 4 task that we are aware of [7] by 7% and 10% relative, respectively. We also showed promising WER gains for a large (open) vocabulary voice search task where we restricted the amount of training data to 100 hours. In this paper, we investigate annealed dropout (AD) within the context of our best ASR training criterion, sequence-level training criterion, and large amounts of training data (>600 hrs). We show that AD-trained Maxout network acoustic models can significantly improve the ST performance of our best ASR systems in both the limited data scenario (Babel limited language pack, LLP, tasks), and the scenario of large amounts of available training data on a voice search task.

2. DROPOUT

The basic dropout training procedure involves, for each new training case, randomly zeroing each dimension of the input to the model (or network layer) with probability p_d, where p_d is the dropout rate. This is equivalent to introducing iid Bernoulli-distributed multiplicative noise into the model, which *masks* each input with probability p_d. This procedure can be viewed as a method of training an *ensemble* of models that share a common set of parameters—each model in the ensemble has a unique input mask associated with it, and as such, utilizes a unique subset of the parameters of the model [2]. Jointly training the parameters of such an ensemble of models implements a powerful form of regularization—each weight is optimized to perform well in the context of the exponential set of models that utilize it. For a log-linear model with inputs $x \in \mathbb{R}^n$ aggregated over a collection of models sharing weights $\{w_{ij}\}$, and each model utilizing a unique mask in the set of all $|\mathcal{M}| = 2^n$ possible binary masks over these shared weights, $m_{:|\mathcal{M}} \in \mathbb{R}^n : m_{j|\mathcal{M}} \in \{0, 1\}$, the geometric average of such a set of exponential models reduces to:

$$\begin{aligned} E_{\mathcal{M}}[\log p(y|x)] &\propto \sum_{\mathcal{M}} p(\mathcal{M}) \log p(y|x, \mathcal{M}) \\ &\propto \sum_{\mathcal{M}} p(\mathcal{M}) \sum_j m_{j|\mathcal{M}} w_{ij} x_j \\ &= \sum_j E_{\mathcal{M}}[m_j] w_{ij} x_j \qquad (1) \end{aligned}$$

where $E_{\mathcal{M}}[m_j] = 1 - p_d$, and p_d is the dropout rate. Therefore at test time the expected output over the geometric mean of the 2^N models being aggregated can be computed by simply turning dropout off, and scaling by the dropout rate utilized during training—a remarkable result. More generally for deep neural networks, no such results hold. However, dropout is effective in practice, particularly

978-1-4673-6997-8/15/$31.00 ©2015 IEEE

for conditionally linear models such as rectified linear (ReLU), Maxout, and Sortout [6] networks.

3. MAXOUT NETWORKS

Maxout networks [5] generalize rectified linear ($\max[0, a]$) units, utilizing non-linearities of the form:

$$s_j = \max_{i \in C(j)} a_i \quad (2)$$

where the activations a_i are based on inner products with the outputs of the layer below: $a_i = \sum_k w_{ik} x_k + b_i$. In the case of activations with unconstrained weights, the sets $C(j) \forall j$ are generally disjoint [5] . Such "pooling" can of course also be overlapping, as is the case for Maxout CNNs [5] and networks layers constrained to have local receptive fields (LRFs) [8], where pooling is done over spatially "local" activations. The units in Order Statistic or Sortout Networks [6] generalize Maxout networks by outputing more general order statistics over such sets of inputs. In this paper we will investigate annealed dropout training predominantly for Maxout networks.

4. ANNEALED DROPOUT

Deterministic annealing is a technique with roots in statistical physics and the maximum entropy principle, and has been applied in machine learning in the context of several non-convex problems, such as expectation-maximization (EM) based learning, and point matching problems, to mitigate against convergence to poor local minima [9–11]. Essentially any regularization parameter can be viewed as a "temperature" parameter, and annealing its value over the course of training will gradually allow for more complex explanations of the data to evolve. Dropout is a powerful regularizer of model complexity, as every weight is constrained to improve the performance of the exponential number of models that share the same parameter, and annealing the dropout rate as a temperature parameter is an effective way to mitigate against the poor solutions. As pointed out in [12], dropout training can be viewed as a Monte Carlo approach that optimizes the expected loss over the ensemble of models formed by all possible masks over node outputs—a Bayesian objective. As a stochastic algorithm annealed dropout does more than gradually increase the theoretical capacity of the network; it also mitigates against the convergence to poor local minima, by ensuring that gradient information is flowing through all parts of the network during training, which can lead to increases in the *realized* capacity of the learned network.

An annealed dropout algorithm has two main components: 1) an annealing schedule that determines the dropout probability for a given epoch, mini-batch, or training case, and 2) the usual dropout procedure, which was already described. In this paper as in [1] we use the following simple annealing schedule:

$$p_d[t] = \max(0, 1 - \frac{t}{N}) p_d[0] \quad (3)$$

to anneal the dropout rate by a constant amount over N steps, unless otherwise specified. More generally variable-rate schedules take the form $p_d[t] = p_d[t-1] + \alpha_t(\theta)$ where $\alpha_t(\theta)$ is an annealing rate parameter that can optionally depend on the current state (or estimate of the state) of the current/auxiliary inputs/parameters θ. Note that the term "annealing" implies that the dropout probability is non-increasing but variable rate schedules (e.g. sample the dropout rate from a current distribution estimate) could also be utilized.

4.1. Interpretation

As discussed previously the dropout procedure implements an aggregation over an exponential number of models, each with a unique mask over the set of weights for a given layer of the network. Annealed dropout realizes a training procedure where the ensemble of models being learned during iteration i is initialized by an ensemble of models with a lower average number of non-zero weights, and higher variance in the number of active weights. This is easily seen given that the probability distribution over n the number of active (not dropped out) units in a layer of units with the same dropout probability is binomial-distributed, and therefore:

$$E[n] = N(1 - p_d) \quad (4)$$

$$Var[n] = N(1 - p_d) p_d \quad (5)$$

Where N is the number of outputs of the layer, and n is the number of "surviving", non-zero outputs.

Note that annealing the dropout rate during stochastic training is related to but different than doing cross-validation to determine the dropout rate. For a log-linear model, which is convex, training to convergence each time the dropout rate is reduced implements a validation search procedure for the dropout rate as a regularization parameter, on the heldout set. For non-convex optimization problems such as neural network training, annealing the dropout rate is more than an (approximate) validation procedure. Annealed dropout biases the learned model toward simple explanations of the data during early training iterations, and gradually increases the capacity of the model to allow more complex explanations to evolve for phenomena that cannot easily be explained. Furthermore, annealed dropout like dropout is a noisy training procedure, which can greatly increase the *realized* capacity of the learned model, again by mitigating against the convergence to pool local optima.

5. EXPERIMENTS ON AURORA 4

The performance of AD was tested extensively on the Aurora 4 robust ASR task in [1], which is a small scale (10 hour), medium vocabulary noise and channel ASR robustness task based on the Wall Street Journal corpus [13]. The most important results from that paper are shown in table 1. These results show that Maxout networks trained using AD consistently outperform those trained using an optimized, fixed dropout rate. Moreover, the best results on the task obtained using AD-trained Maxout and Sortout LRF networks outperform the best published previous result on the task [7] by 10% and 7% relative, respectively, despite the lack of use of any features for robustness.

6. BABEL EXPERIMENTS

To further investigate the performance of AD-trained Maxout networks in the context of limited data and the use of sequence-training criterion we have begun performing experiments on Babel limited language pack (LLP) tasks (see [15] for details). As sequence training can be considered a "fine-tuning" phase of model training, two immediate questions are: 1) How to best utilize an AD-trained CE models during ST?, and, 2) Do the CE gains obtained using AD survive ST?

	WER (%)				
Network	A	B	C	D	AVG
NAT [14]	5.4	8.3	7.6	18.5	12.4
JNAT [7]	4.5	7.4	8.1	16.5	11.1
ReLU, #H=1414	4.9	8.7	8.2	16.9	11.9
Maxout,#H=1024	4.4	8.7	7.8	16.9	11.8
AD Maxout , #H=1024	4.3	7.7	7.0	15.6	10.8
ReLU CNN #H=1414	4.9	8.1	7.3	15.5	11.0
Maxout CNN, #H=1024	4.6	8.2	7.2	15.2	10.9
AD Maxout CNN, #H=1024	4.0	7.8	6.7	14.9	10.5
ReLU LRF, #H=1414	4.7	8.3	7.5	16.1	11.3
Maxout LRF #H=1024	4.2	7.8	7.0	15.6	10.8
AD Maxout LRF, #H=1024	4.2	7.4	6.5	14.8	10.3

Table 1. Word error rate (WER) as a function of network type on the Aurora 4 task. All networks utilize 7 hidden layers, and have roughly the same number of parameters. All Maxout networks utilize 2 linear filters per hidden unit. The number of hidden units per layer for each network is indicated. The best published results on Aurora 4 that we are aware of have also been included. Networks trained using annealed dropout (AD) are noted–all others results use their optimal fixed dropout rate. Please consult [1] for further details.

		WER (%)	
Network	AD scheme	CE	ST
Sigmoid DNN	D=0	80.2	77.4
Maxout DNN	CE: AD, ST: D=0	79.3	75.9
Maxout DNN	CE: AD ST: D=fixed	79.3	75.7
Maxout DNN	CE:AD, ST:AD	79.3	77.0

Table 2. Word error rate (WER) as a function of model and training method for various the DNNs trained on the Babel's Tamil LLP. The WERs for both cross-entropy (CE) and the subsequently sequence-trained (ST) models are given. Several strategies for utilizing an AD-trained CE model during ST are considered. AD methods reduce the dropout rate by 0.02 per epoch.

W.r.t. the first question, there are, roughly speaking, 3 options: 1) re-normalize the model based on the final CE dropout rate and turn dropout off, 2) leave the dropout rate fixed at the value inferred during CE training, and 3) continue to anneal the dropout rate as ST proceeds. Table 2 compares the performance of several AD-trained maxout networks on the Tamil LLP task to a highly tuned baseline system based on the sigmoid non-linearity. All DNN acoustic models have 5 hidden layers of 1024 hidden units and a softmax output layer. The input features to all networks were computed based on 9 adjacent frames of 40 dimensional features, which were speaker-adapted using FMLLR. All networks are initialized with layer-wise discriminative pre-training. After the pre-training, the models were subjected to up to 30 iterations of cross-entropy (CE) training followed by up to 30 iterations of Hessian-free (HF) sequence training based on the state-level minimum Bayesian error (sMBR). It is worthy of note that other researchers at IBM have trained ReLU networks on these features and have failed to produce WER gains over this baseline at the ST level. Table 2 summarizes the results we obtained. These results suggest that it may be better to keep the final dropout rate identified during CE training fixed during ST, at least in the case of severely limited data and less researched ASR languages like Tamil. Note that for the ST:AD scheme, the best model is found at iteration 10, whereas the other models improve until iteration 30. Further iterations with a fixed dropout rate would further improve this result, however the ST:D=0 model is already at 76.4% WER at iteration 10.

	WER (%)	
Network	CE	ST
Sigmoid CNN	64.6	62.6
AD Maxout CNN	63.2	59.7
D=0.375 Maxout CNN (Best D)	63.2	60.8

Table 3. Word error rate (WER) as a function of model and training method for Babel's Hatian-Creole limited language pack (LLP). The WERs for both cross-entropy (CE) and the subsequently sequence-trained (ST) models are given. During AD the dropout rate was reduced from 0.5 to 0.24 over the first 13 CE epochs.

To begin to investigate the general importance of AD during ST in limited data scenarios, we next compared AD-trained Maxout CNNs to Maxout CNNs trained with a fixed dropout rate and a highly-tuned baseline CNN system for Hatian-Creole. The baseline CNN model has two convolutional layers followed by four fully connected feedforward layers. All hidden layers utilize the sigmoid activation function. The input features to the first conv. layer are 40-dim. log-Mel features with VTLN and their deltas and double deltas. Eleven frames of temporal context are then spiced. There are 128 feature maps in the first conv. layer. each with 9x9 local receptive fields (LRFs), which results in 32x3-dim feature maps. These are max-pooled by a 3x1 non-overlapping windows to produce 11x3-dim. outputs. There are 256 feature maps in the second conv. layer each with 4x3 LRFs, which results in 8x1-dim feature maps. Following the second convolutional layer are four fully connected feedforward layers, each containing 1024 units. The training of the CNN is composed of up to 30 iterations of CE training followed by 20 iterations of HF sMBR sequence training. The Maxout networks tested utilize 2 filters per Maxout unit and the same topology in terms of number of feature maps and hidden units. Table 3 depicts the results. In contrast with our results on Aurora 4, the AD-trained system is only on-par with best fixed-dropout system at the CE level. This presumably because the annealing schedule was not tuned for the task. In any case, we have avoided a grid-search over dropout rate by using a pre-defined AD schedule tuned on Aurora 4. Both systems significantly outperform the highly tuned baseline, and the AD-trained ST system outperforms the best fixed-dropout ST Maxout system, perhaps because better lattices result. Significant WER gains over the baseline system have recently been obtained by re-aligning with the output ST model and re-training, and by ASR-specific data augmentation [16]. These techniques could further improve the results.

7. EXPERIMENTS: VOICE SEARCH

The presented results on the Aurora 4 and Babel limited language pack (LLP) tasks show that even in data-limited situations, annealed dropout can deliver further gains over a fixed dropout training strategy. We now investigate the use of Maxout networks trained with annealed dropout for commerical open voice search (OVS) in the data-plenty scenario.

7.1. Experimental Setup

Experiments were conducted using an IBM internal open voice search (OVS) US English task for ASR. The data consists of mobile search queries and messaging tasks consisting of 633.8 hours of manually transcribed data. for a total of 711K utterances. This data set was split in two subsets: a training set is made of 622.3 hours of data (698K utterances) randomly chosen from the full OVS set, and a heldout set is composed of the remaining 11.5 hours (13K utterances). Our decoding set Test-15K is made solely of 19.2 hours of mobile search queries from 14.9K unique speakers (for 17.8K utterances). A random subset of 5K speakers called Test-5K is about 6.6 hour long (6071 utterances), and used moslty for acoustic weight tuning. We report WERs on both sets.

All DNN models were first trained based on a cross-entropy (CE) objective using SGD for up to 30 epochs, and additionally discriminatively pre-trained for one epoch as each hidden layer was added. Sequence training was done using an evolved version of the training system described in [15], which utilizes a Hessian-free (HF) training procedure similar to that described in [16]. We utilized a modified procedure called Dynamic Stochastic Average Gradient with HF optimization (DSAG-HF) as introduced in [17] that displays faster convergence than regular HF-ST. All models operate on 31-dim log Mel filterbanks, which are stacked with their 1st, and 2nd time derivatives, and spliced over +-5 frames to yield 1023- dim input features. All DNNs are trained to predict 9000 context dep. acoustic states derived from a context dependency tree that was learned from our training data.

7.2. Results

Table 4 summarizes the DNN topologies that were investigated. Note that all models have roughly the same number of parameters.

Table 5 summarizes the performance of several DNN models on a 100 hour subset of the OVS training data. All models were trained for 30 CE and 10 ST iterations based on forced alignments derived from the 622 hr baseline system. The AD-trained Maxout model outperforms the baseline model, and slightly outperforms the optimal fixed dropout rate Maxout model, while alleviating the need to search for the best fixed dropout rate.

Table 6 depicts CE and ST results for several models trained on the full 622 training set of this OVS task. Looking at the results, we can see that the 0.5% absolute gain that AD Maxout LRFs had over the baseline Sigmoid system is essentially maintained when we move from from CE to an ST criterion, and from the test-5K (7hrs) dataset to the larger test-15K (19 hrs) dataset. So far no other network topologies have been able to improve this baseline system, despite promising indications when working with smaller amounts of data. In contrast, these results show that AD-trained maxout networks *are* capable of improving on this highly tuned traditional DNN baseline system by a significant margin.

8. DISCUSSION

In this paper we have shown empirical evidence that Maxout networks, when trained with annealed dropout, can improve ASR systems trained using sequence-level discriminative criterion, and LVCSR systems trained on large amounts of training data. While these results are encouraging, several important research directions remain. Currently we are conducting experiments to quantitatively study the effects of the dropout annealing schedule on performance, with a particular focus on big data regimes–so far results suggest that WER performance is quite insensitive to the annealing schedule, but that annealing the dropout rate to zero over the first 1000-5000 hours of data encountered by SGD produces good results. Another important and directly related research question is how the “search” for the annealing schedule (for both the dropout and learning rate) should be carried out—methods that can dynamically decide or average over candidate models e.g. based on a dynamic tree of learning/dropout rates and their associated model parameters are of great interest to us, but so far such approaches have not delivered speed or performance gains. Such pursuits are particularly intriguing in the scenario of asynchronous stochastic model updating algorithms.

Topology	Non-linearity	#H x #L + P
Baseline	Sigmoid	2K x 5 + 100 (Linear)
Maxout	Maxout, 2 filters/unit	1.4K x 4 + 512
Maxout LRF	Maxout, 2 filters/unit	LRF + 1.4K x 3 + 512
Maxout LRFP	Maxout, 2 filters/unit	LRF + 1.4K x 4 + 100

Table 4. Topology of models investigated for OVS task. The number of hidden layers (#L), units per hidden layer (#H), size of the ’pinch’ layer (P) immediately before the output layer, and the network nonlinearity are as specified. All models predict 9000 context-dependent acoustic states. The local receptive field (LRF) used in the first layer of topology M2 utilizes 40 9x9 filters per time-frequency position. All networks have roughly the same number of parameters.

		WER (%)	
Topology	Dropout	5K CE	5K ST
Baseline	No	13.0	11.7
Maxout	D=0.2 (best D)	12.9	11.5
Maxout	AD (0.5 → 0 over 20 epochs)	12.6	11.3

Table 5. Word error rate (WER) as a function of model topology and training method. All models were trained on 100 hrs of voice search data. Sequence-trained (ST) models we initialized with their corr. (CE) models. Acoustic weights and log priors were tuned on the Test 5K data (7 hrs). Models were trained using annealed dropout (AD) where specified.

		WER (%)		
Topology	Dropout	5K CE	5K ST	15K ST
Baseline	No	11.5	10.1	10.7
Maxout	AD	11.2	9.8	10.3
Maxout LRF	AD	11.0	9.7	10.2
Maxout LRFP	AD	11.1	9.7	10.1

Table 6. Word error rate (WER) as a function of model topology and training method. All models were trained on 622 hrs of open voice search (OVS) data. Sequence-trained (ST) models we initialized with the corr. (CE) models (see text for details). Acoustic weights and log priors were tuned on the Test 5K data (7 hrs), which is a subset of the Test 15K (20 hrs). AD-trained models had their dropout rate annealed to zero over the first 3 epochs.

9. REFERENCES

[1] Steven Rennie, Vaibhava Goel, and Samuel Thomas, "Annealed dropout training of deep networks," in *Spoken Language Technology (SLT), IEEE Workshop on*. IEEE, 2014.

[2] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[3] Geoffrey E Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[4] George E Dahl, Tara N Sainath, and Geoffrey E Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8609–8613.

[5] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.

[6] Steven Rennie, Vaibhava Goel, and Samuel Thomas, "Deep order statistic networks," in *Proc. of the IEEE Workshop on Spoken Language Technology (SLT)*, 2014.

[7] Arun Narayanan and DeLiang Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Proc. of the IEEE ICASSP*, 2014.

[8] Quoc V Le, "Building high-level features using large scale unsupervised learning," in *Proc. of IEEE ICASSP*, 2013.

[9] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.

[10] A.L. Yuille, P. Stolorz, and J. Utans, "Statistical physics, mixtures of distributions, and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 334–340, 1994.

[11] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2000, vol. 2, pp. 44–51.

[12] Sida Wang and Christopher Manning, "Fast dropout training," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 118–126.

[13] N. Parihar and J. Picone, "Aurora working group: DSP frontend and LVCSR evaluation au/384/02," *Tech. Rep., Inst. for Signal and Information Processing,Mississippi State University*, 2002.

[14] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE ICASSP*, 2013.

[15] Jia Cui, Xiaodong Cui, Bhuvana Ramabhadran, Janice Kim, Brian Kingsbury, Jonathan Mamou, Lidia Mangu, Michael Picheny, Tara N Sainath, and Abhinav Sethy, "Developing speech recognition systems for corpus indexing under the iarpa babel program," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6753–6757.

[16] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury, "Data augmentation for deep convolutional neural network acoustic modeling," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6753–6757.