APPENDICES

A DATASETS AND TASKS

Below we provide a summary of datasets used in the experiments.

NLP tasks The NLP datasets information is summarized in Table 1.

- **MRPC** (Microsoft Research Paraphrase Corpus) (Dolan & Brockett, 2005) is a corpus of sentence pairs extracted from online news sources. Human annotation indicates whether the sentences in the pair are semantically equivalent. We report accuracy and F1 score.
- **SST-2** (The Stanford Sentiment Treebank) (Socher et al., 2013) is a task to determine the sentiment of a given sentence. This corpus contains sentences from movie reviews and their sentiment given by human annotations. We use only sentence-level labels, and predict positive or negative sentiment.
- **QNLI** is a converted dataset from the Stanford Question Answering Dataset (Rajpurkar et al., 2016) which consists of question-paragraph pairs. As in (Wang et al., 2018), this task is to predict whether the context sentence selected from the paragraph contains the answer to the question.
- **QQP** (Quora Question Pairs dataset) (Iyer et al., 2017) contains question pairs from the question-answering website Quora. Similar to MRPC, this task is to determine whether a pair of questions are semantically equivalent. We report accuracy and F1 score.

ASR tasks The speech datasets are summarized in Table 4.

- **TIMIT** (Garofolo et al., 1993) consists of speech from American English speakers, along with the corresponding phonemical and lexical transcription. It is widely used for acoustic-phonetic classification and ASR tasks. Its training set, validation set and test set are 3.2 hours, 0.15 hours, 0.15 hours long, respectively.
- **WSJ** (Wall Street Journal corpus) (Paul & Baker, 1992) contains read articles from the Wall Street Journal newspaper. Its training, validation and test set are 80 hours, 1.1 hours and 0.7 hours long, respectively.
- **Librispeech** (Panayotov et al., 2015) is a large-scale (1000 hours in total) corpus of 16 kHz English speech derived from audiobooks. We choose the subset train-clean-100 (100 hours) as our training data, dev-clean (2.8 hours) as our validation set and test-clean (2.8 hours) as our test set.

Vision tasks The vision datasets information is summarized in Table 6.

- **MNIST** (LeCun et al., 1998) contains $60,000$ training images and $10,000$ testing 28×28 pixel images of hand-written digits. It is a 10-class image classification task.
- **CIFAR-10** (Krizhevsky & Hinton, 2009) consists of $50,000$ 32×32 pixel training images and $10,000$ 32×32 pixel test images in 10 different classes. It is a balanced dataset with $6,000$ images of each class.
- **ImageNet** (Russakovsky et al., 2015) is an image dataset with 1000 classes, and about 1.28 million images as training set. The sizes of its validation and test set are $50,000$ and $10,000$, respectively. All images we use are in 224×224 pixels.

B HYPER-PARAMETER SETTINGS

We give the implementation toolkits and specific hyper-parameter settings to help reproduce our results, and list the epochs needed for training with the square loss and the cross-entropy (CE) loss. The data processing is following the standard methods. For NLP tasks, it is the same as in (Wang et al., 2018), and for ASR tasks, it is the same as in (Watanabe et al., 2018). For vision tasks, we are following the default ones given in the implementation of the corresponding papers.

B.1 HYPER-PARAMETERS FOR NLP TASKS

The implementation of BERT is based on the PyTorch toolkit (Wolf et al., 2019). The specific script we run is https://github.com/huggingface/transformers/blob/master/examples/text-classification/run_glue.py, and we use the bert-base-cased model for fine-tuning. LSTM+Attention and LSTM+CNN are implemented based on the toolkit released by (Lan & Xu, 2018). The specific hyper-parameters used in the experiments are in Table 10. As there are many hyper-parameters, we only list the key ones, and all other parameters are the default in the scripts.

Table 10: Hyper-parameters for NLP tasks

Model	Task	Batch size	max_seq length	Learning rate w/		Epochs training w/	
				square loss	CE	square loss	CE
BERT	MRPC	32	128	5e-5	2e-5	5	3
	SST-2	32	128	2e-5	2e-5	3	3
	QNLI	32	128	2e-5	2e-5	3	3
	QQP	32	128	2e-5	2e-5	3	3
LSTM+Attention	MRPC	64	80	2e-4	1e-4	25	20
	QNLI	32	$sent\_len^*$	1e-4	1e-4	20	20
	QQP	64	120	1e-4	1e-4	30	30
LSTM+CNN	MRPC	64	80	2e-4	1e-4	20	20
	QNLI	32	$sent\_len^*$	8e-5	1e-4	20	20
	QQP	32	120	1e-3	1e-3	20	20

* The max sequence length equals the max sentence length of the training set.

B.2 HYPER-PARAMETERS FOR ASR TASKS

The implementation of ASR tasks is based on the ESPnet (Watanabe et al., 2018) toolkit, and the specific code we use is the run.sh script under the base folder of each task, which is https://github.com/espnet/espnet/tree/master/egs/?/asr1, where '?' can be 'timit', 'wsj', and 'librispeech'. The specific hyper-parameters are following the ones in the configuration file of each task, which is under the base folder. We list the files which give the hyper-parameter settings for acoustic model training in Table 11.

Table 11: Hyper-parameters for ASR tasks

Model	Task	Hyper-parameters	Epochs training w/	
			square loss	CE
Attention+CTC	TIMIT	conf/train.yaml♮	20	20
VGG+BLSTMP	WSJ*	conf/tuning/train_rnn.yaml	15	15
VGG+BLSTM	Librispeech	conf/tuning/train_rnn.yaml♢	30	20

* For WSJ, we use the language model given by https://drive.google.com/open?id=1Az-4H25uwnEFa4lENc-EKiPaWXaijcJp. ♮ We set mtlalpha=0.3, batch-size=30. ♢ We set elayers=4, as we use 100 hours training data.

B.3 HYPER-PARAMETERS FOR VISION TASKS

The implementation of these models are based on the open source toolkits. For TCNN and EfficientNet, we use the open source implementation given by (Bai et al., 2018) and (Tan & Le, 2019), respectively. For Wide ResNet, we are based on the open source PyTorch implementation https://github.com/xternalz/WideResNet-pytorch (W-ResNet). For ResNet-50, our experiments are based on the Tensorflow toolkit https://github.com/tensorflow/tpu/tree/master/models/official/resnet (ResNet) implemented on TPU. The hyper-parameter settings for our vision experiments are in Table 12.

Table 12: Hyper-parameters for vision tasks

Model	Task	Hyper-parameters	Epochs training w/	
			square loss	CE
TCNN	MNIST[♮]	the default in (Bai et al., 2018)	20	20
Wide-ResNet	CIFAR-10	the default in W-ResNet, except wide-factor=20	200	200
ResNet-50	ImageNet	the default in ResNet, for square loss, learning rate=0.3	168885*	112590*
EfficientNet	ImageNet	the default in EfficientNet-B0 of (Tan & Le, 2019)	218949*	218949*

[♮] We are doing the permuted MNIST task as in Bai et al. (2018).
* We give the training steps as in the original implementations.

C EXPERIMENTAL RESULTS ON VALIDATION AND TRAINING SETS

We report the results for validation set of NLP tasks in Table 13 for accuracy and Table 14 for F1 scores.

Table 13: NLP results on validation set, accuracy

Model	Task	train with square loss (%)	train with cross-entropy (%)	square loss w/ same epochs as CE (%)
BERT (Devlin et al., 2018)	MRPC	**85.3**	85.0	**85.3**
	SST-2	91.2	**91.5**	91.2
	QNLI	**90.8**	90.7	**90.8**
	QQP	**90.8**	90.7	90.6
LSTM+Attention (Chen et al., 2017)	MRPC	**76.5**	74.8	75.3
	QNLI	**79.7**	**79.7**	**79.7**
	QQP	**86.0**	85.5	**86.0**
LSTM+CNN (He & Lin, 2016)	MRPC	**76.0**	73.3	**76.0**
	QNLI	**76.8**	**76.8**	**76.8**
	QQP	84.0	**85.3**	84.0

Table 14: NLP results on validation set, F1 scores

Model	Task	train with square loss (%)	train with cross-entropy (%)	square loss w/ same epochs as CE (%)
BERT (Devlin et al., 2018)	MRPC	89.5	**89.6**	89.5
	QQP	**87.5**	87.4	87.4
LSTM+Attention (Chen et al., 2017)	MRPC	**83.7**	83.3	83.5
	QQP	**82.1**	81.7	**82.1**
LSTM+CNN (He & Lin, 2016)	MRPC	**82.6**	81.4	**82.6**
	QQP	77.4	**80.2**	77.4

The validation set results of the ASR tasks are in Table 15.

Table 15: ASR results on validation set, error rate

Model	Task	train with square loss (%)	train with cross-entropy (%)	square loss w/ same epochs as CE (%)
Attention+CTC (Kim et al., 2017)	TIMIT (PER)	**18.1**	18.3	**18.1**
	TIMIT (CER)	**30.4**	31.4	**30.4**
VGG+BLSTMP (Moritz et al., 2019)	WSJ (WER)	**8.5**	8.8	**8.5**
	WSJ (CER)	**3.9**	4.0	**3.9**
VGG+BLSTM (Moritz et al., 2019)	Librispeech (WER)	**9.3**	10.7	9.9
	Librispeech (CER)	**9.4**	11.1	10.2

We report the training result for NLP tasks in Table 16 for accuracy and F1 score in Table 17.

Table 16: NLP results on training and test set, accuracy

Model	Task	train with square loss (%)		train with cross-entropy (%)		square loss w/ same epochs as CE (%)	
		Train	Test	Train	Test	Train	Test
BERT (Devlin et al., 2018)	MRPC	99.7	83.8	99.9	82.1	99.6	83.6
	SST-2	98.6	94.0	99.2	93.9	98.6	93.9
	QNLI	98.0	90.6	97.5	90.6	98.0	90.6
	QQP	96.2	88.9	98.0	88.9	96.2	88.8
LSTM+Attention (Chen et al., 2017)	MRPC	94.6	71.7	84.9	70.9	93.2	71.5
	QNLI	87.7	79.3	90.8	79.0	87.7	79.3
	QQP	93.7	83.4	91.5	83.1	93.7	83.4
LSTM+CNN (He & Lin, 2016)	MRPC	98.3	73.2	92.5	69.4	98.3	72.5
	QNLI	92.8	76.0	90.7	76.0	92.8	76.0
	QQP	91.3	84.3	95.7	84.4	91.3	84.3

Table 17: NLP results on training and test set, F1 scores

Model	Task	train with square loss (%)		train with cross-entropy (%)		square loss w/ same epochs as CE (%)	
		Train	Test	Train	Test	Train	Test
BERT (Devlin et al., 2018)	MRPC	99.8	88.1	99.9	86.7	99.7	88.0
	QQP	94.5	70.9	97.2	70.7	94.5	70.7
LSTM+Attention (Chen et al., 2017)	MRPC	96.1	80.9	89.5	80.6	94.7	80.7
	QQP	91.9	62.6	89.2	62.3	91.9	62.6
LSTM+CNN (He & Lin, 2016)	MRPC	98.8	81.0	94.5	78.2	98.8	81.0
	QQP	88.0	60.3	94.2	60.5	88.0	60.3

D OUR RESULTS COMPARED WITH THE ORIGINAL WORK

We list our results for the models trained with the cross-entropy (CE) loss and compare them to the results reported in the literature or the toolkits in Table 18. As we observe, our results are comparable to the original reported results.

Table 18: Training with the cross-entropy loss, our results and the reported ones

Model	Task	Our CE result	CE result in the literature
BERT*	MRPC (acc./F1)	85.0/89.6	85.29/89.47 (Wolf et al., 2019)
	SST-2 (acc.)	91.5	91.97 (Wolf et al., 2019)
	QNLI (acc.)	90.7	87.46 (Wolf et al., 2019)
	QQP (acc./F1)	90.7/87.4	88.40/84.31 (Wolf et al., 2019)
LSTM+Attention			N/A
LSTM+CNN			N/A
Attention+CTC	TIMIT (PER)	20.7	20.5 (Watanabe et al., 2018)
	TIMIT (CER)	32.7	33.7 (Watanabe et al., 2018)
VGG+BLSTMP	WSJ (WER)	5.4	5.3 (Watanabe et al., 2018)
	WSJ (CER)	2.6	2.4 (Watanabe et al., 2018)
VGG+BLSTM	Librispeech (WER)	10.8	N/A
	Librispeech (CER)	11.0	N/A
TCNN	MNIST (acc.)	98.0	97.2 (Bai et al., 2018)
Wide-ResNet	CIFAR-10 (acc.)	96.5	96.11 (Zagoruyko & Komodakis, 2016)
ResNet-50	ImageNet (acc./Top-5 acc.)	76.1/93.0	76.0/93.0 (Tan & Le, 2019)
EfficientNet	ImageNet (acc./Top-5 acc.)	77.2/93.4	77.3/93.5 (Tan & Le, 2019)

* The implementation in (Wolf et al., 2019) is using bert-base-uncased model, we are using bert-base-cased, which will result in a little difference. Also, as they didn't give test set results, here for BERT, we give the results of validation set.

The models marked with 'N/A' in Table 18 do not have comparable results reported in the literature. Specifically, LSTM+Attention and LSTM+CNN models for NLP tasks are implemented based on

the toolkit released by (Lan & Xu, 2018), where they did not show results on MRPC and QNLI. The QQP results are not comparable with ours as they were using a different test set, while we are using the standard test set same as in (Wang et al., 2018). The VGG+BLSTM model for Librispeech dataset is based on ESPnet toolkit (Watanabe et al., 2018). Due to computational resources limitations, we only use train-clean-100 (100 hours) as training data and 1000 unigram based dictionary for acoustic model training, while they use 1000 hours of training data with at least 2000 unigram dictionary.

E REGULARIZATION TERMS

We give the regularization term of each task in Table 19. 0 means we didn't add regularization term. For WSJ, check the details at line 306 of https://github.com/espnet/espnet/blob/master/espnet/nets/pytorch_backend/rnn/decoders.py.

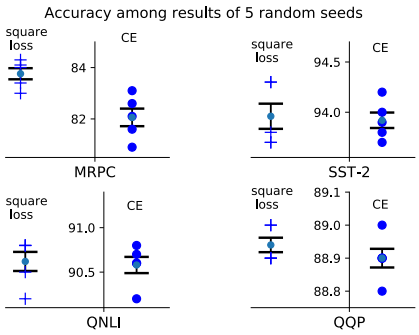
Table 19: Regularization term for each task

Model	Task	dropout*	batch norm	Regularization Term
BERT	MRPC/SST-2/QNLI/QQP	0.1	N	0
LSTM+Attention	MRPC/QNLI/QQP	0.5	N	0
LSTM+CNN	MRPC/QNLI/QQP	0.0	N	0
Attention+CTC	TIMIT	0.0	N	0
VGG+BLSTMP	WSJ	0.0	N	label smoothing based
VGG+BLSTM	Librispeech	0.0	N	0
TCN	MNIST	0.05	N	0
Wide-ResNet	CIFAR-10	0.0	N	0
ResNet-50	ImageNet	0.0	Y	$\frac{10^{-4}}{2}\sum_{i=1}^{n} \boldsymbol{w_i^2}$
EfficientNet	ImageNet	0.0	Y	$\frac{10^{-5}}{2}\sum_{i=1}^{n} \boldsymbol{w_i^2}$

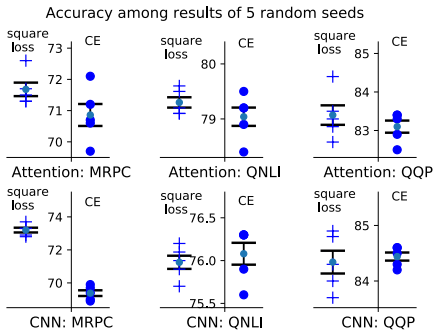
* For dropout, 0.0 means have not apply dropout.

F VARIANCE OF ACCURACY AMONG DIFFERENT RANDOM SEEDS

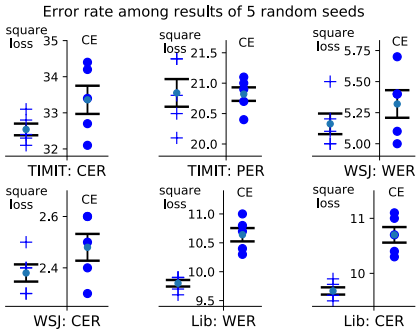
Figure 3 gives the error bar of 5 runs corresponding to 5 different random seeds, along with the results for each inidividual run. In the left of each subfigure is the result of training with the square loss, while in the right is result of the cross-entropy. As can be seen in Figure 3, using the square loss has better accuray/error rate and smaller variance in NLP and ASR tasks, which indicates that training with the square loss for those classification tasks is statistically better.



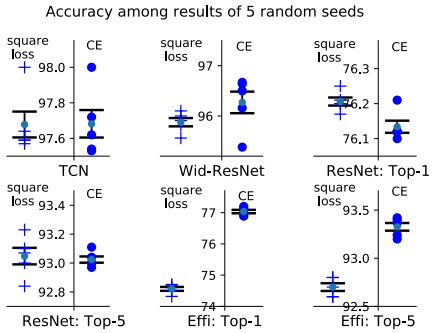
(a) NLP: BERT



(b) NLP: LSTM+Attention & LSTM+CNN



(c) ASR



(d) Vision

Figure 3: Accuracy/error rate variance of results among 5 random seeds