Appendix: Cell-aware Stacked LSTMs for Modeling Sentences

Model	# Encoder Layers	Bidirectional	# MLP Layers	p_w	p_c
2-layer CAS-LSTM	2		1	0.10	0.10
2-layer Bi-CAS-LSTM	2	✓	2	0.15	0.15
3-layer CAS-LSTM	3		2	0.15	0.20
3-layer Bi-CAS-LSTM	3	✓	2	0.15	0.15

Table 1: Hyperparameters for SNLI models.

Model	# Encoder Layers	Bidirectional	# MLP Layers	p_w	p_c	L_2 weight
2-layer CAS-LSTM	2		1	0.10	0.10	0.0025
2-layer Bi-CAS-LSTM	2	✓	2	0.15	0.20	0.0020
3-layer CAS-LSTM	3		2	0.10	0.15	0.0025
3-layer Bi-CAS-LSTM	3	✓	2	0.15	0.20	0.0020

Table 2: Hyperparameters for SNLI models.

Model	# Encoder Layers	Bidirectional	# MLP Layers	p_w	p_c
CAS-LSTM	2		1	0.10	0.10
Bi-CAS-LSTM	2	✓	1	0.15	0.20

Table 3: Hyperparameters for Quora Question Pairs models.

1 Experimental Settings

1.1 Weight Initialization

Weight matrices for recurrent connections are initialized according to the orthogonal initialization scheme [7]. All other weight matrices are initialized using the scheme proposed by He et al. [2], except the weights for the last fully-connect layer which are initialized by sampling from the uniform distribution $\mathcal{U}(-0.005, 0.005)$. Bias vectors are initialized to zero.

1.2 SNLI

For all models, 300D 840B GloVe pretrained vectors [6] are used as word embedding and not updated during training. Adam optimizer [4] is used for training, and the learning rate is annealed according to cosine schedule [5] with the initial learning rate of 0.001. For all models, we added the L_2 norm of the parameters to the classification loss with the factor of 0.002. The dimensions of encoder states and MLP hidden layers are set to 300 and 1024 respectively. Batch normalization [3] and dropout [8] is applied to the input and each layer output of the MLP. Dropout is also applied to

Model	# Encoder Layers	Bidir.	Encoder Dim.	MLP Hidden Dim.	p_w	p_c	L_2 weight
CAS-LSTM	2		300	300	0.5	0.5	0.010
Bi-CAS-LSTM	2	✓	150	300	0.5	0.5	0.005

Table 4: Hyperparameters for SST-2 models.

Model	# Encoder Layers	Bidir.	Encoder Dim.	MLP Hidden Dim.	p_w	p_c	L_2 weight
CAS-LSTM	2		300	300	0.4	0.4	0.005
Bi-CAS-LSTM	2	✓	300	300	0.5	0.5	0.005

Table 5: Hyperparameters for SST-5 models.

word embeddings, and we denote the drop probability of word embeddings by p_w and that of MLP input and layer outputs by p_c. The maximum length of each sentence is 35, and words beyond the sentence boundary are discarded. Each minibatch is composed of 128 data samples.

The number of encoder LSTM layers, MLP hidden layers, and dropout probabilities (p_w, p_c) differ among models. We summarize values of these hyperparameters in Table 1.

1.3 MutiNLI

For MultiNLI models we mostly use the same setting as for the SNLI models, but in this time L_2 weights differ from model to model and the maximum length is set to 55. Table 2 summarizes the hyperparameters for MultiNLI models.

1.4 Quora Question Pairs

Again, models for the Quera Question Pairs dataset are identical to those used for SNLI and MultiNLI. All models use the L_2 weight of 0.002. The hyperparameters are listed in Table 3.

1.5 SST

As in other experiments 840B 300D GloVe vectors are used, but in this time they are updated during training. Models are trained using AdaDelta algorithm [9] instead of Adam. Table 4 and 5 describe hyperparameters used for SST-2 and SST-5 models respectively.

1.6 IWSLT 2014

We used the fairseq [1] for experiments. `lstm_wiseman_iwslt_de_en` is used as the base architecture, and we implemented the CAS-LSTM counterpart. We set the number of LSTM layers to 2 and selected the dropout probability p from $\{0.1, 0.2, 0.3\}$, and set $p = 0.1$ for the base architecture and $p = 0.2$ for the CAS-LSTM architecture.

References

[1] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *ICML*, pages 1243–1252, 2017.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[5] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, Toulon, France, 2017.

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[7] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *ICLR*, Banff, Canada, 2014.

[8] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *NIPS*, pages 2377–2385, 2015.

[9] Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *CoRR, cs.LG/1212.5701v1*, 2012.