Empirical Investigation of Optimization Algorithms in Neural Machine Translation

Parnia Bahar, Tamer Alkhouli, Jan-Thorsten Peter, Christopher Jan-Steffen Brix, Hermann Ney

bahar@i6.informatik.rwth-aachen.de

29th May, 2017
EAMT 2017, Prague, Czech Republic

Human Language Technology and Pattern Recognition
Computer Science Department, RWTH Aachen University

Introduction

- **Neural Machin Translation (NMT) trains a single, large neural network reading a sentence and generates a variable-length target sequence**
- **Training an NMT system involves the estimation of a huge number of parameters in a non-convex scenario**
- **Global optimality is given up and local minima in the parameter space are considered sufficient**
- **Choosing an appropriate optimization strategy can not only obtain better performance, but also accelerate the training phase of neural networks and brings higher training stability**

Related work

- [Im & Tao$^+$ 16] try to show the performance of optimizers in the investigation of loss surface for image classification task
- [Zeyer & Doetsch$^+$ 17] investigate various optimization methods for acoustic modeling empirically
- [Dozat 15] compares different optimizers in language modeling
- [Britz & Goldie$^+$ 17] study a massive analysis of NMT hyperparameters aiming for better optimization being robust to the hyperparameter variations
- [Wu & Schuster$^+$ 16] utilize the combination of Adam and a simple Stochastic Gradient Descend (SGD) learning algorithm

This Work - Motivation

- A study of the most popular optimization techniques used in NMT
- Averaging the parameters of a few best snapshots from a single training run leads to improvement [Junczys-Dowmunt & Dwojak^{+} 16]
- An open question concerning training problem
- Either the model or the estimation of its parameters is weak

This work

- Empirically investigate the behavior of the most prominent optimization methods to train an NMT
- Investigate the combinations that seek to improve optimization
- Addressing three main concerns:
  - translation performance
  - convergence speed
  - training stability
- First, how well, fast and stable different optimization algorithms work
- Second, how a combination of them can improve these aspects of training

Neural Machine Translation

- **Given a source $\mathrm{f} = f_1^J$ and a target $\mathrm{e} = e_1^I$ sequence, NMT [Sutskever & Vinyals$^+$ 14, Bahdanau & Cho$^+$ 15] models the conditional probability of target words given the source sequence**
- **The NMT training objective function is to minimize the cross-entropy over the S training samples $\left\{\left\langle \mathrm{f}^{(s)}, \mathrm{e}^{(s)} \right\rangle\right\}_{s=1}^{S}$**

$$J(\theta) = \sum_{s=1}^{S} \sum_{i=1}^{I^{(s)}} \log p(\mathrm{e}_i{}^{(s)} | \mathrm{e}_{<i}{}^{(s)}, \mathrm{f}^{(s)}; \theta)$$

Stochastic Gradient Descent (SGD) [Robbins & Monro 51]

- SGD updates a set of parameters, θ
- g_t represents the gradient of the cost function J
- η is called the learning rate, determining how large the update is
- Tunning of the learning carefully

Algorithm 1 : Stochastic Gradient Descent (SGD)

1: $g_t \leftarrow \nabla_{\theta_t} J\left(\theta_t\right)$
2: $\theta_{t+1} \leftarrow \theta_t - \eta g_t$

Adagrad
[Duchi & Hazan+ 11]

- The shared global learning rate η is divided by the l_2-norm of all previous gradients, n_t
- Different learning rates for every parameter
- Larger updates for the dimensions with infrequent changes and smaller updates for those that have already large changes
- n_t in the denominator is a positive growing value which might aggressively shrink the learning rate

Algorithm 2 : Adagrad

1: $g_t \leftarrow \nabla_{\theta_t} J(\theta_t)$
2: $n_t \leftarrow n_{t-1} + g_t^2$
3: $\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{\sqrt{n_t} + \epsilon} g_t$

RmsProp
[Hinton & Srivastava+ 12]

- **Instead of storing all the past squared gradients from the beginning of the training, a decaying weight of squared gradients is applied**

Algorithm 3 : RmsProp

1: $g_t \leftarrow \nabla_{\theta_t} J\left(\theta_t\right)$
2: $n_t \leftarrow \nu n_{t-1} + (1-\nu) g_t^2$
3: $\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{\sqrt{n_t} + \epsilon} g_t$

Adadelta
[Zeiler 12]

- Takes the decaying mean of the past squared gradients
- The squared parameter updates, s_t, is accumulated in a decaying manner to compute the final update
- Since $\Delta\theta_t$ is unknown for the current time step, its value is estimated by the r_t of parameter updates up to the last time step

Algorithm 4 : Adadelta

1: $g_t \leftarrow \nabla_{\theta_t} J(\theta_t)$
2: $n_t \leftarrow \nu n_{t-1} + (1-\nu) g_t^2$
3: $r(n_t) \leftarrow \sqrt{n_t + \epsilon}$
4: $\Delta\theta_t \leftarrow \frac{-\eta}{r(n_t)} g_t$
5: $s_t \leftarrow \nu s_{t-1} + (1-\nu)\Delta\theta_t^2$
6: $r(s_{t-1}) \leftarrow \sqrt{s_{t-1} + \epsilon}$
7: $\theta_{t+1} \leftarrow \theta_t - \frac{r(s_{t-1})}{r(n_t)} g_t$

Adam
[Kingma & Ba 15]

- The decaying average of the past squared gradients n_t
- Stores a decaying mean of past gradients m_t
- First and second moments

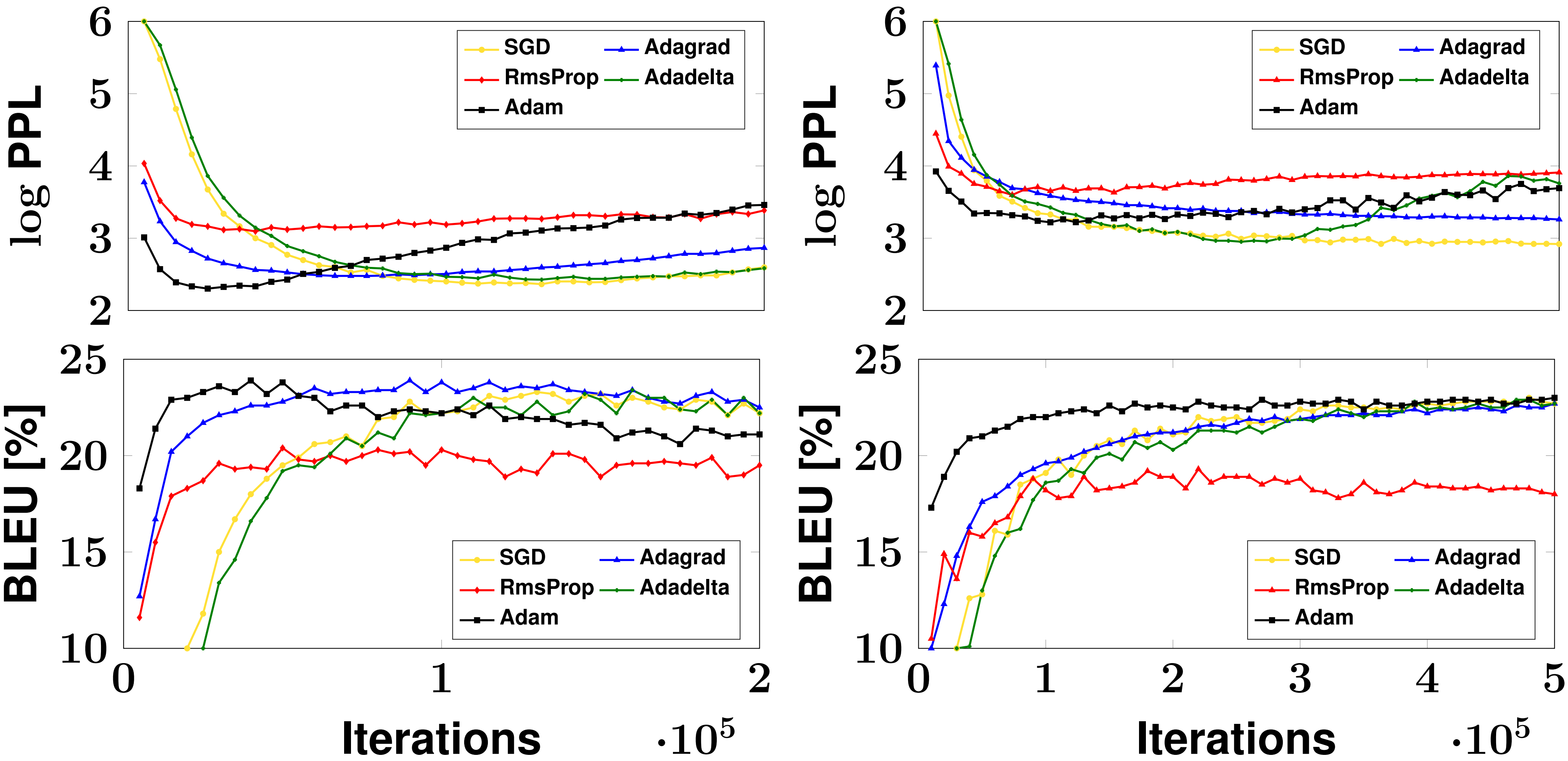
Algorithm 5 : Adam

1: $g_t \leftarrow \nabla_{\theta_t} J\left(\theta_t\right)$
2: $n_t \leftarrow \nu n_{t-1} + (1-\nu) g_t^2$
3: $\hat{n}_t \leftarrow \frac{n_t}{1-\nu^t}$
4: $m_t \leftarrow \mu m_{t-1} + (1-\mu) g_t$
5: $\hat{m}_t \leftarrow \frac{m_t}{1-\mu^t}$
6: $\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{\sqrt{\hat{n}_t}+\epsilon} \hat{m}_t$

Experiments

- Two translation tasks, the WMT 2016 En$\rightarrow$Ro and WMT 2015 De$\rightarrow$En
- NMT model follows the architecture by [Bahdanau & Cho^{+} 15]
- joint-BPE approach [Sennrich & Haddow^{+} 16]
- Evaluate and save the models on validation sets every 5k iterations for En$\rightarrow$Ro and every 10K iterations for De$\rightarrow$En
- The models are trained with different optimization methods
  - the same architecture
  - the same number of parameters
  - identically initialized by the same random seed

Analysis - Individual Optimizers



(a) En→Ro **(b) De→En**

Figure: $\log$ **PPL and BLEU score of all optimizers on the val. sets.**

Combination of Optimizers

- A fast convergence at the beginning, then reducing the learning rate
- take advantage of methods which accelerate the training and afterwards switch to the techniques with more control on the learning rate
- Starting the training with any of the five considered optimizers, pick the best model, then continue training the network
  1. Fixed-SGD: simple SGD algorithm with a constant learning rate. Here, we use a learning rate of 0.01
  2. Annealing: annealing schedule in that the learning rate of optimizer is halved after every sub-epoch
- Reaching an appropriate region in the parameter space and it is a good time to slow down the training. By means of finer search, the optimizer has better chance not to skip good local minima

Results

		En→Ro	De→En
	Optimizer	`newsdev16` BLEU	`newsdev11+12` BLEU
1	SGD	23.3	22.8
2	+ Fixed-SGD	24.7 (+1.4)	23.8 (+1.0)
3	+ Annealing-SGD	24.8 (+1.5)	24.1 (+1.3)
4	Adagrad	23.9	22.6
5	+ Fixed-SGD	24.2 (+0.3)	22.4 (-0.2)
6	+ Annealing-SGD	24.3 (+0.4)	22.9 (+0.3)
7	+ Annealing-Adagrad	24.6 (+0.7)	22.6 (0.0)
8	Adadelta	23.2	22.9
9	+ Fixed-SGD	24.5 (+1.3)	23.8 (+0.9)
10	+ Annealing-SGD	24.6 (+1.4)	24.0 (+1.1)
11	+ Annealing-Adadelta	24.6 (+1.4)	24.0 (+1.1)
12	Adam	23.9	23.0
13	+ Fixed-SGD	26.2 (+2.3)	24.5 (+1.5)
14	+ Annealing-SGD	26.3 (+2.4)	24.9 (+1.9)
15	+ Annealing-Adam	26.2 (+2.3)	25.4 (+2.4)

Table: Results in BLEU[%] on val. sets.

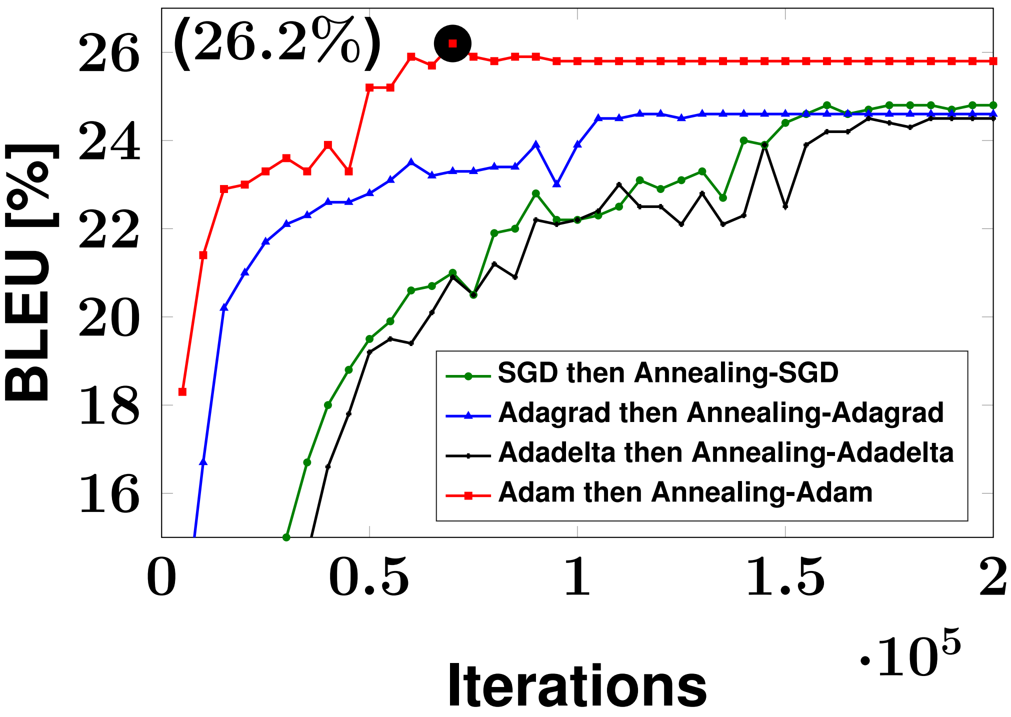
Results - Performance

	Optimizer	En→Ro `newstest16`	De→En `newstest15`
1	SGD	20.3	26.1
2	+ Annealing-SGD	22.1	27.4
3	Adagrad	21.6	26.2
4	+ Annealing-Adagrad	21.9	25.5
5	Adadelta	20.5	25.6
6	+ Annealing-Adadelta	22.0	27.6
7	Adam	21.4	25.7
8	+ Annealing-Adam	23.0	29.0

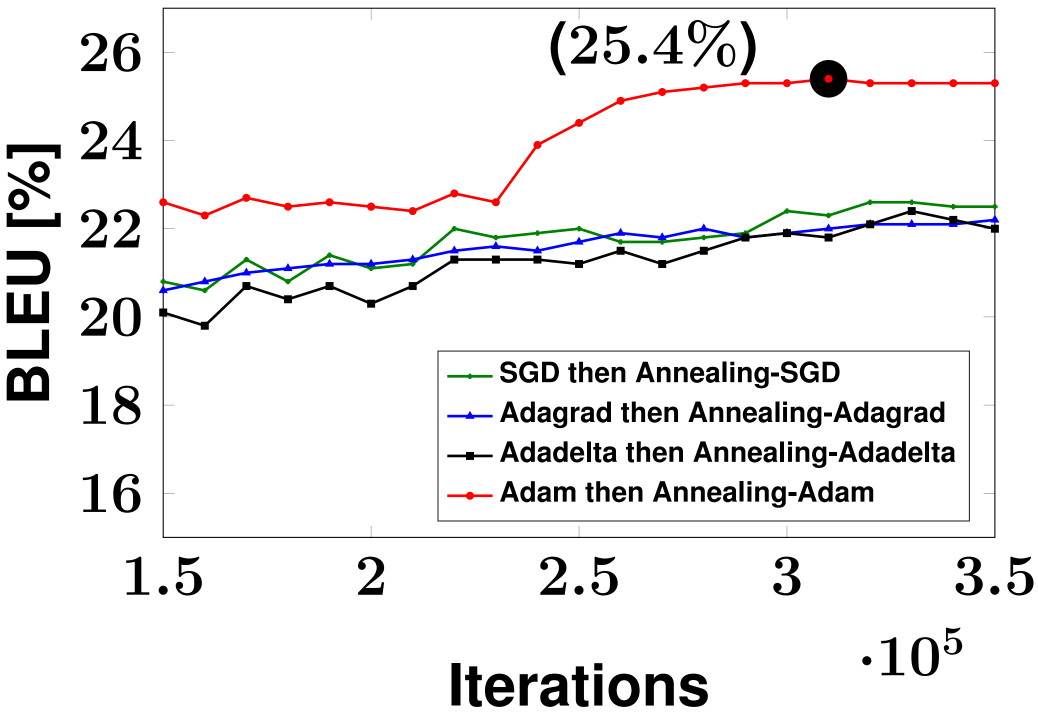
Table: Results measured in BLEU[%] on the test sets.

- Shrinking the learning steps might lead to a finer search and prevent stumbling over a local minimum
- Adam followed by Annealing-Adam gains the best performance

Results - Convergence Speed



(a) En→Ro



(b) De→En

Figure: BLEU score of the best combination on the val. sets.

- Faster convergence in the training by Adam followed Annealing-Adam

Results - Training Stability

		De→En newstest15	
	Optimizer	Best Model	Averaged-best
1	SGD	26.1	27.4
2	+ Annealing-SGD	27.4	27.2
3	Adagrad	26.2	26.0
4	+ Annealing-Adagrad	25.5	25.5
5	Adadelta	25.6	27.4
6	+ Annealing-Adadelta	27.6	27.4
7	Adam	25.7	28.9
8	+ Annealing-Adam	29.0	29.0

Table: Results measured in BLEU[%] for best and averaged-best models on the test sets.

- Pure Adam training is less regularized and stumbles on good cases
- Adam+Annealing-Adam is more regularized, leading to less varieties

Conclusion

- Practically analyzed the performance of common gradient-based optimization methods in NMT
- Ran alone or followed by the variations differing in the handling of the learning rate
- The quality of the models in terms of BLEU scores as well as the convergence speed and robustness against stochasticity have been investigated on two WMT translation tasks
- Apply Adam followed by Annealing-Adam
- Experiments done on WMT 2016 En$\rightarrow$Ro and WMT 2015 De$\rightarrow$En show that the mentioned technique leads to 1.6% BLEU improvements on `newstest16` for En$\rightarrow$Ro, and 3.3% BLEU on `newstest15` for De$\rightarrow$En
- It results to faster convergence as well as the training stability

Thank you for your attention

Parnia Bahar, Tamer Alkhouli, Jan-Thorsten Peter, Christopher Jan-Steffen Brix, Hermann Ney

`<surname>@i6.informatik.rwth-aachen.de`

Analysis - Combination

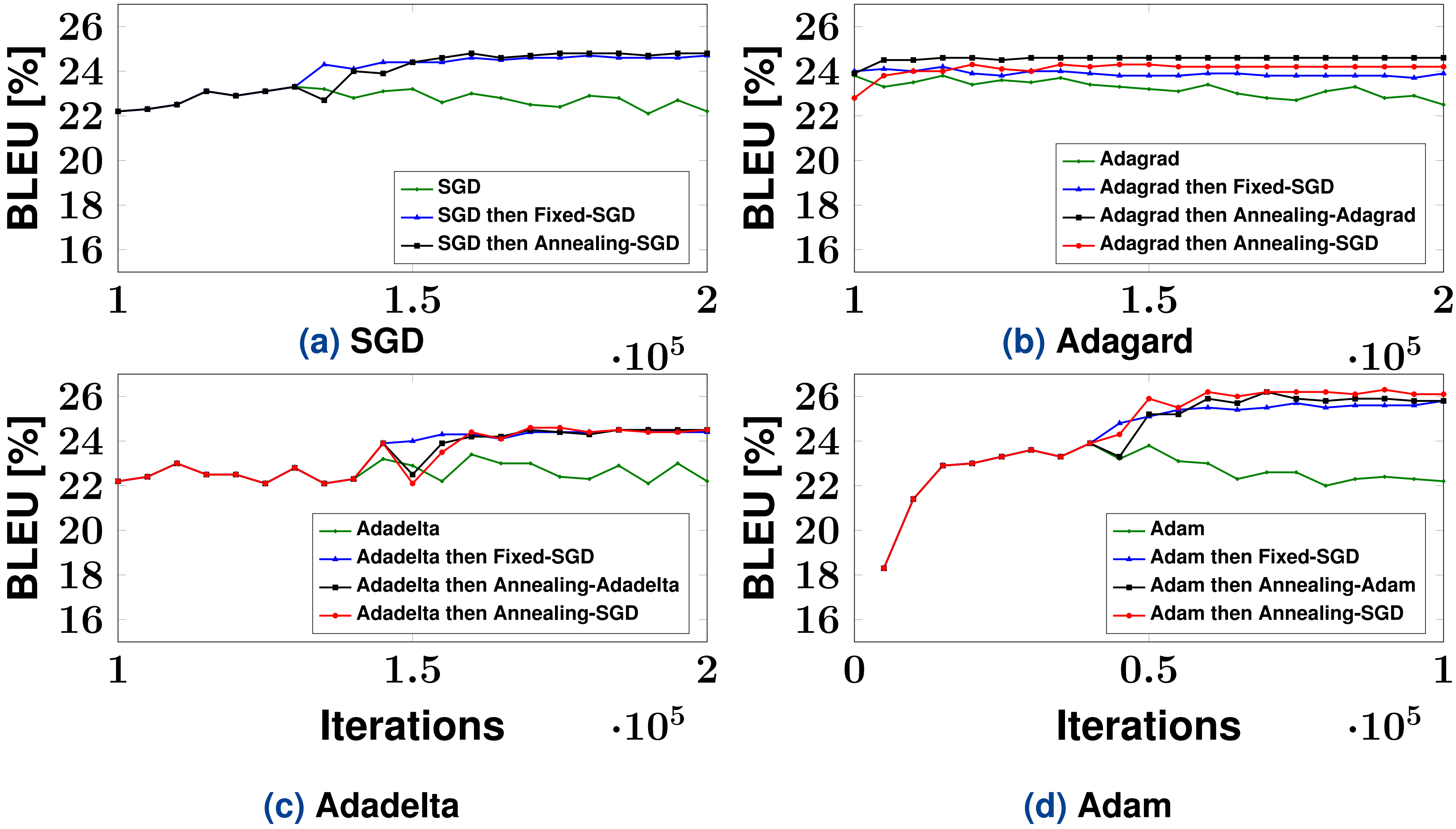


Figure: BLEU of optimizers followed by the combinations on the val. set for En→Ro.

Analysis - Combination

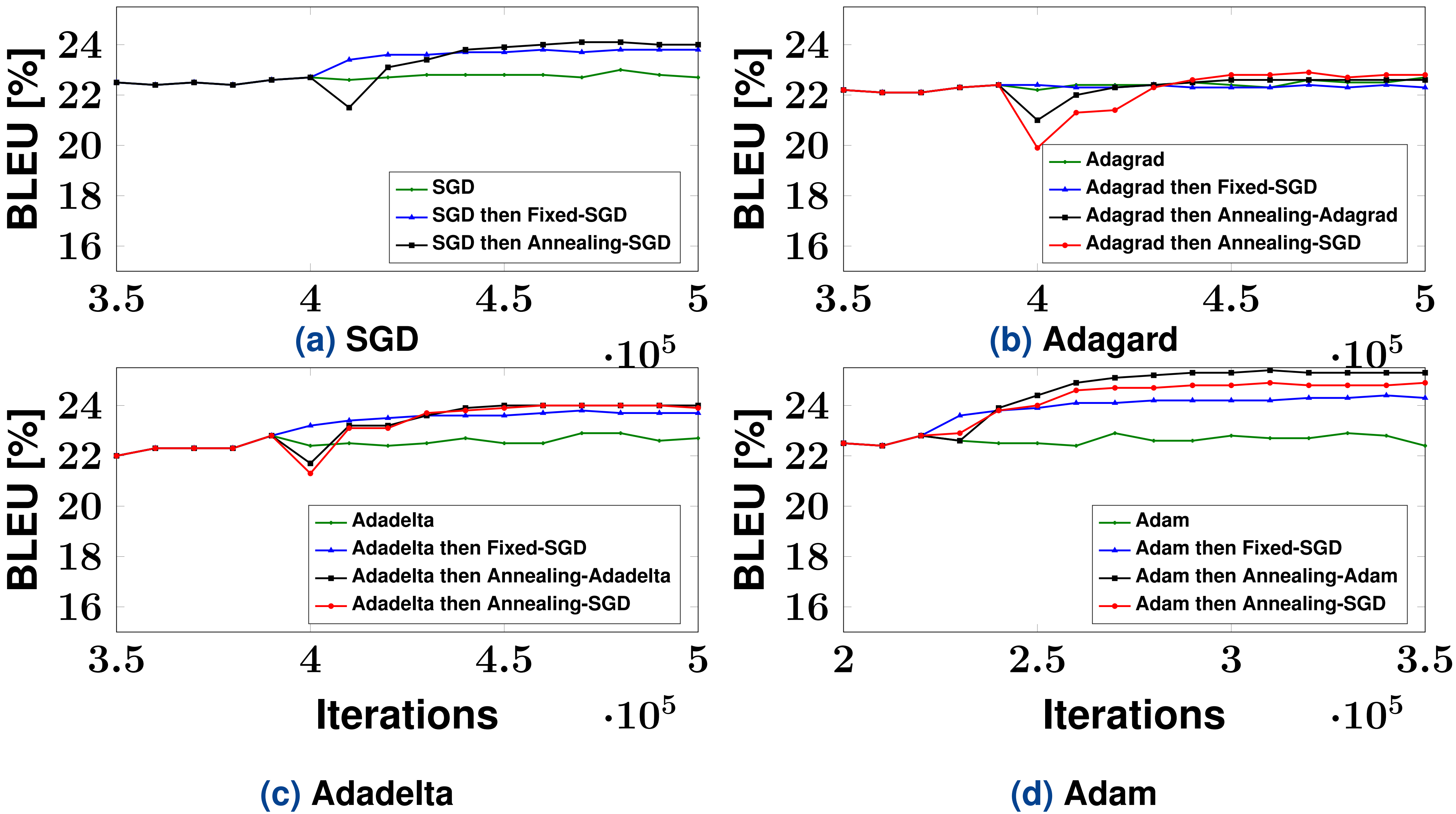


Figure: BLEU of optimizers followed by the combinations on the val. set for De→En.

Reference

- **D. Bahdanau, K. Cho, Y. Bengio.**
  Neural machine translation by jointly learning to align and translate.
  CoRR, Vol. abs/1409.0473, 2015.

- **F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, Y. Bengio.**
  Theano: new features and speed improvements.
  Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

- **D. Britz, A. Goldie, T. Luong, Q. Le.**
  Massive exploration of neural machine translation architectures.
  arXiv preprint arXiv:1703.03906, Vol., 2017.

- K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio.
  On the properties of neural machine translation: Encoder-decoder approaches.
  In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Qatar*, pp. 103–111, 2014.
- J. H. Clark, C. Dyer, A. Lavie, N. A. Smith.
  Better hypothesis testing for statistical machine translation: Controlling for optimizer instability.
  In *49th Annual Meeting of the Association for Computational Linguistics*, pp. 176—181, USA, 2011.
- T. Dozat.
  Incorporating nesterov momentum into adam.
  Technical report, 2015.

- J. C. Duchi, E. Hazan, Y. Singer.
  Adaptive subgradient methods for online learning and stochastic optimization.
  Journal of Machine Learning Research, Vol. 12, pp. 2121–2159, 2011.
- M. A. Farajian, R. Chatterjee, C. Conforti, S. Jalalvand, V. Balaraman, M. A. Di Gangi, D. Ataman, M. Turchi, M. Negri, M. Federico.
  Fbk's neural machine translation systems for iwslt 2016.
  In *Proceedings of the ninth International Workshop on Spoken Language Translation, USA*, 2016.
- I. Goodfellow, Y. Bengio, A. Courville.
  ***Deep Learning*.**
  MIT Press, 2016.
- G. Hinton, N. Srivastava, K. Swersky.
  Lecture 6a overview of mini–batch gradient descent.
  Coursera Lecture slides https://class.coursera.org/neuralnets-2012-001/, Vol., 2012.

- D. J. Im, M. Tao, K. Branson.
  An empirical analysis of deep network loss surfaces.
  CoRR, Vol. abs/1612.04010, 2016.
- S. Jean, O. Firat, K. Cho, R. Memisevic, Y. Bengio.
  Montreal neural machine translation systems for wmt'15.
  In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT 2015, Portugal*, pp. 134–140, 2015.
- M. Junczys-Dowmunt, T. Dwojak, R. Sennrich.
  The AMU-UEDIN submission to the WMT16 news translation task: Attention-based NMT models as feature functions in phrase-based SMT.
  In *Proceedings of the First Conference on Machine Translation, WMT 2016, Germany*, pp. 319–325, 2016.
- D. P. Kingma, J. Ba.
  Adam: A method for stochastic optimization.
  CoRR, Vol. abs/1412.6980, 2015.

- **T. Luong, H. Pham, C. D. Manning.**
  Effective approaches to attention-based neural machine translation.
  In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 2015*, pp. 1412–1421, 2015.

- **B. Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski, Y. Bengio.**
  Blocks and fuel: Frameworks for deep learning.
  Vol., 2015.

- **K. Papineni, S. Roukos, T. Ward, W.-J. Zhu.**
  Bleu: a Method for Automatic Evaluation of Machine Translation.
  In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, USA, 2002.

- **H. Robbins, S. Monro.**
  A stochastic approximation method.
  The annals of mathematical statistics, Vol., pp. 400–407, 1951.

- **S. Ruder.**
  An overview of gradient descent optimization algorithms.
  CoRR, Vol. abs/1609.04747, 2016.
- **R. Sennrich, B. Haddow, A. Birch.**
  Neural machine translation of rare words with subword units.
  In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Germany*, 2016.
- **M. Snover, B. Dorr, R. Schwartz, L. Micciulla, J. Makhoul.**
  A Study of Translation Edit Rate with Targeted Human Annotation.
  In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pp. 223–231, USA, 2006.
- **I. Sutskever, O. Vinyals, Q. V. Le.**
  Sequence to sequence learning with neural networks.
  In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Canada*, pp. 3104–3112, 2014.

- Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, Klaus et al.
  Google's neural machine translation system: Bridging the gap between human and machine translation.
  CoRR, Vol. abs/1609.08144, 2016.
- M. D. Zeiler.
  ADADELTA: an adaptive learning rate method.
  CoRR, Vol. abs/1212.5701, 2012.
- A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, H. Ney.
  A comprehensive study of deep bidirectional LSTM rnns for acoustic modeling in speech recognition.
  CoRR, Vol. abs/1606.06871, 2017.