Minimization of learning errors: case of invariance of geometric transformations in license plate recognition

Hassan T. Kajila[1], Jordan Masakuna[2], Pascal Sungu, and Ilunga Godwill

[1] Université Nouveaux Horizons, Lubumbashi, DR Congo,
infos@unhorizons.org,
WWW home page: https://www.unhorizons.org/
[2] Université de Kinshasa, Département des Sciences Informatiques, Kinshasa, DR Congo

Abstract. Nowadays, a huge number of cameras are deployed exclusively for surveillance. The contents of images and videos are often interpreted by human operators. This makes content tracking and analysis prohibitively expensive, not to mention the mistakes that human fatigue and carelessness can introduce.
This work aims to propose a method, based on artificial intelligence, which allows computer systems to derive meaningful information from digital images, with the lowest possible cost. As part of this work, the aim is to perform automatic license plate recognition using optimizers based on stochastic gradient descent.
Therefore, the main objective of this present study is to examine the different optimizers, finally to determine the best able to recognize license plates, despite possible geometric transformations. In order to reduce the constraints in the detection of license plates as to the position of the camera.
Experiments on several license plates that undergo a slight geometric transformation, in image, made it possible to validate the performance obtained with this approach.

Key words: Supervised learning, Optimisation, Stochastic gradient descent, Visual Geometry Group (VGG), Automatic License Plate Recognition, Invariance of geometric transformations.

1 Introduction

In this paper, it is a question of approaching the use of the algorithms of numerical optimization, precisely of minimization of the errors. They will be applied to machine learning that will allow computers and computer systems to derive meaningful information from digital images.
Indeed, it is the recognition of vehicle license plates using a classifier and optimizers of the family of stochastic gradient descent implemented in a convolutional neural network (CNN). The result obtained will be used to measure the efficiency of the optimizers with respect to the constraint of invariance to the geometric transformations.
Therefore for each algorithm, the efficiency of geometric transformation invariance will be examined and compared the score (accuracy, loss) for different optimizers.

2 Result & discussion

2.1 CNN VGG Model

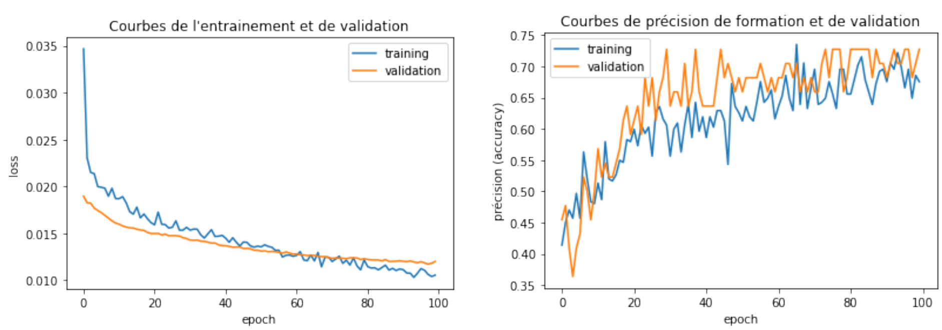
Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 6, 6, 512)	14714688
flatten (Flatten)	(None, 18432)	0
dropout (Dropout)	(None, 18432)	0
dense (Dense)	(None, 256)	4718848
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 4)	260
Total params: 19,474,948		
Trainable params: 4,760,260		
Non-trainable params: 14,714,688		

Table 1: The information the neural network builds with VGG-16.

2.2 Optimizer evaluation

- **SGD** The SGD optimizer minimizes the model parameter with a score of 67.5% accuracy and about 1% error. And on the validation data, the accuracy is 72.7% and an error of 1.2%. The test result reveals an error of 0.73% and an accuracy of 79.3%.

Training	**Validation**	**Test**
loss: 0.0105	val loss: 0.0120	test loss: 0.00732
accuracy: 0.6755	val accuracy: 0.7273	test accuracy: 0.7931



(a) loss : perte

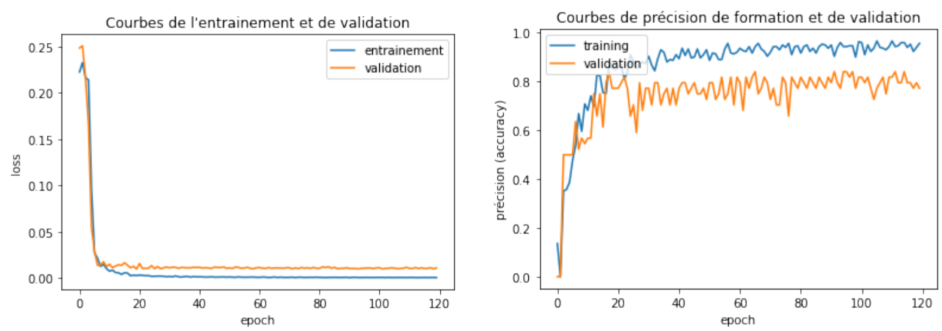
(b) accuracy : précision

Fig. 1: Accuracy and loss graph for SGD.

- **RMSprop** The model minimized with the RMSprop gives a score of 95.7% accuracy, displays a deviation of about 28.2% compared to the SGD, and 0.044%. The error and accuracy rate of RMSpro in validation is 1% and 77.2% respectively. The RMSprop optimizer also shows a lower error percentage than the SGD.

The convergence of RMSprop is rather accentuated because the minimization stabilizes after 80 steps and reaches the minimum value at the 110th step.

Training	**Validation**	**Test**
`loss: 4.4475e-04`	`val loss: 0.0109`	`test loss : 0.005589`
`accuracy: 0.9570`	`val accuracy: 0.7727`	`test accuracy : 0.8505`



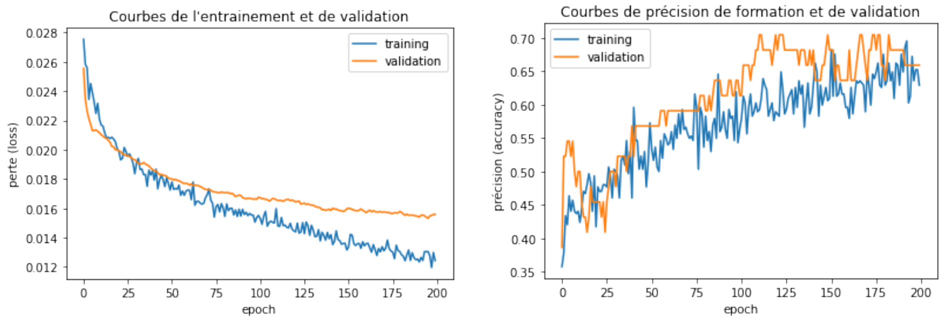
(a) loss : perte

(b) accuracy : précision

Fig. 2: Accuracy and loss graph for RMSprop.

- **AdaGrad** AdaGrad scored poorly compared to the previous optimizer, RMSprop. An error rate of 1.2% and accuracy of 62.9% is the score of the AdaGrad optimizer. After 120 steps, which is the maximum step set for the tests, the minimization still does not stabilize. The minimum values are reached at the level of the 80*th* step.

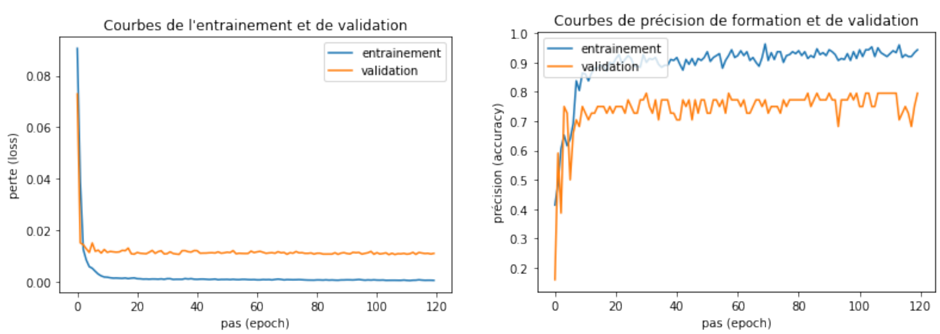
Training	**Validation**	**Test**
`loss: 0.0124`	`val loss: 0.0156`	`test loss: 0.0094`
`accuracy: 0.6291`	`val accuracy: 0.6591`	`test accuracy: 0.5747`



(a) loss : perte (b) accuracy : précision

Fig. 3: Accuracy and loss graph for Adagrad.

Training	Validation	Test
loss: 8.6163e-04	val loss: 0.0110	test loss : 0.00526
accuracy: 0.9238	val accuracy: 0.8182	test accuracy : 0.8390



(a) loss : perte (b) accuracy : précision

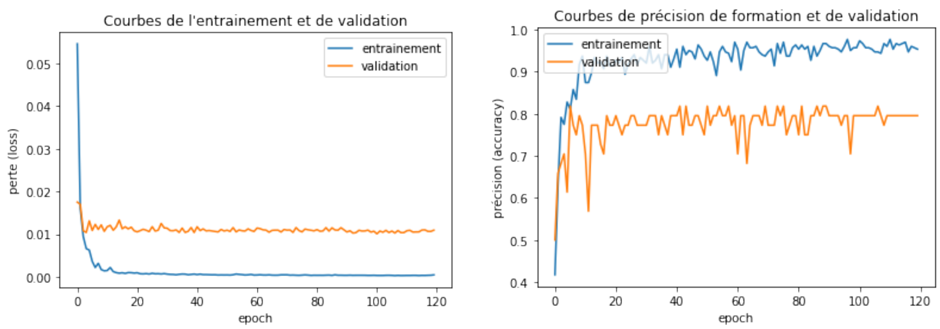
Fig. 4: Accuracy and loss graph for Adam.

- **Adam** The Adam optimizer forms a model with 92.3% accuracy and an error of 0.086%, a bit higher compared to RMSprop. The validation score is 81.8% accuracy and 1.1% error. After the tests, 83.9% accuracy and 0.05% error is the observed score. RMSprop minimizes the cost better and shows a nice score compared to Adam.

- **Nadam** Nadam, (Nesterov-accelerated Adaptive Moment Estimation), is an extension of the Adam optimizer to which we add the accelerated Nesterov gradient (NAG) [41].

Nadam shows an attractive performance (from training, validation and testing point of view) compared to other optimizers studied in this work. It gives a result of 95.3% accuracy and an error of 0.04%. Nadam accelerates the speed of convergence in a drastic way, it stabilizes at 30 steps and reaches the minimum value in only 40 steps.

Training	Validation	Test
loss: 4.6549e-04	val loss: 0.0109	test loss : 0.00479
accuracy: 0.9536	val accuracy: 0.7955	test accuracy : 0.9080



(a) loss : perte (b) accuracy : précision

Fig. 5: Accuracy and loss graph for Nadam

RMSprop gives very low error in training compared to other optimizers, in testing phase Adam reveals lower error than RMSprop. The optimizer which has a good average, in the 3 phase, is Nadam with very good precision.
This work admits the Nadam optimizer as an algorithm that best minimizes errors, in the context of ALPR, compared to the other optimizers studied. Below is a list of the best minimizing optimizers, ranked in descending order, for the case of the license plate recognition problem.

N°	Training	Validation	Test
1. RMSprop	0.0004447	0.0109	0.005597
2. Nadam	0.00046549	0.0109	0.00479
3. Adam	0.00086163	0.0110	0.00526
4. Adagrad	0.0124	0.0156	0.0094
5. SGD	0.0105	0.0120	0.00732

	Training	Validation	Test
Nadam	95.36%	79.55%	90.80%
Adam	92.58%	81.82%	83.90%
RMSprop	95.70%	77.27%	85.05%

Fig. 6: The plates are recognized.

Fig. 7: The plates recognized despite the slight geometric transformation.

References

1. T Syed Akheel, V Usha Shree, and S Aruna Mastani. Stochastic gradient descent linear collaborative discriminant regression classification based face recognition. *Evolutionary Intelligence*, pages 1–15, 2021.
2. E. Alpaydin. *Introduction to Machine Learning*. Adaptive computation and machine learning. MIT Press, 2010.
3. Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.
4. Vincent Barra Antoine Cornuéjols, Laurent Michet. *Apprentissage artificiel : Deep leaning, concepts et algorithmes*. Eyrolles, 3rd edition, 2018.
5. P. Benner, M. Bollhöfer, D. Kressner, C. Mehl, and T. Stykel. *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory: Festschrift in Honor of Volker Mehrmann*. Springer International Publishing, 2015.
6. Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, first edition, 2006.
7. Anna Sergeevna Bosman, Andries Engelbrecht, and Mardé Helbig. Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions. *Neurocomputing*, 400:113–136, 2020.
8. Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.
9. Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.
10. Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.
11. Léon Bottou et al. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.
12. Christopher Burges, Robert Ragno, and Quoc Le. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems*, 19, 2006.
13. Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.
14. IBM Corporation. What is computer vision?, 2017.
15. F. Coulombeau, G. Debeaumarché, B. David, F. Dorra, S. Dupont, and M. Hochart. *Mathématiques MPSI-PCSI: Programme 2013 avec algorithmique en Scilab*. Cap Prépa. Pearson, 2013.
16. Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia*, pages 21–49. Springer, 2008.
17. R.B. Darlington and A.F. Hayes. *Regression Analysis and Linear Models: Concepts, Applications, and Implementation*. Methodology in the Social Sciences. Guilford Publications, 2016.
18. Natarajan Deepa, B Prabadevi, Praveen Kumar Maddikunta, Thippa Reddy Gadekallu, Thar Baker, M Ajmal Khan, and Usman Tariq. An ai-based intelligent system for healthcare analysis using ridge-adaline stochastic gradient descent classifier. *The Journal of Supercomputing*, 77:1998–2017, 2021.
19. Oliver K Ernst. Stochastic gradient descent learning and the backpropagation algorithm. *University of California, San Diego, La Jolla, CA, Tech. Rep*, 2014.
20. Kary Främling. Scaled gradient descent learning rate. *Reinforcement Learning With Light-Seeking Robot, Proceedings of ICINCO*, pages 1–8, 2004.
21. Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999.

22. A. Géron. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, 2017.
23. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* Adaptive Computation and Machine Learning series. MIT Press, 2016.
24. Google Brain Team. Tensorflow.
25. Guido van Rossum. Python.
26. F.E. Harrell and F.E.H. jrl. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis.* Graduate Texts in Mathematics. Springer, 2001.
27. Heartex Labs. Labelimg.
28. Intel & Willow Garage. Opencv.
29. JAIDED AI. Easyocr.
30. Daniel Kirsch Judith Hurwitz. *Machine Learning For Dummies.* John Wiley and Sons, Inc., ibm limited edition edition, 2018.
31. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
32. Donald E. Knuth. Computer Programming as an Art. *Communications of the ACM*, 17(12):667–673, 1974.
33. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
34. Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv preprint arXiv:1908.06281*, 2019.
35. Agnes Lydia and Sagayaraj Francis. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci*, 6(5):566–568, 2019.
36. N. Matloff. *Statistical Regression and Classification: From Linear Models to Machine Learning.* Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2017.
37. Praneeth Netrapalli. Stochastic gradient descent and its variants in machine learning. *Journal of the Indian Institute of Science*, 99(2):201–213, 2019.
38. Jorge Nocedal and Stephen J Wright. *Numerical optimization*, volume 35. 1999.
39. Arnu Pretorius, Elan Van Biljon, Steve Kroon, and Herman Kamper. Critical initialisation for deep signal propagation in noisy rectifier neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.
40. Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
41. Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
42. D. Sarkar, R. Bali, and T. Sharma. *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems.* Apress, 2017.
43. Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. *Model assisted survey sampling.* Springer Science & Business Media, 2003.
44. Vahid Mirjalili Sebastien Raschka. *Python Machine Learning and Deep Learning, with sckit-learn and Tensorflow.* Packt, 2nd edition, 2017.
45. Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.
46. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

47. Srikanth Tammina. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, 9(10):143–150, 2019.
48. Rob GJ Wijnhoven and PHN de With. Fast training of object detection using stochastic gradient descent. In *2010 20th International Conference on Pattern Recognition*, pages 424–427. IEEE, 2010.
49. Wei Yu, Kuiyuan Yang, Yalong Bai, Tianjun Xiao, Hongxun Yao, and Yong Rui. Visualizing and comparing alexnet and vgg using deconvolutional layers. In *Proceedings of the 33 rd International Conference on Machine Learning*, 2016.