Design and Implementation of a Deep-Learning Course to Address AI Industry Needs

Bastian Beggel and Johannes Schmitt

Kaiserslautern University of Applied Sciences
bastian.beggel@hs-kl.de, johannes.schmitt@hs-kl.de

Abstract. Industry in Europe and worldwide currently has high interest in deep learning, but many medium-sized and smaller companies fear significant investments in specialized hardware and shortage of qualified personal. The goal of our work is to address these two issues. The University of Applied Sciences in Kaiserslautern is starting to offer a deep learning course in its new computer science master program. The course specifically addresses industry needs by focusing on the practical applications of deep learning in the fields of computer vision and natural language processing. We enable our students to perform deep learning projects by teaching hands-on technical expertise and programming tools that are required to do so using cloud systems. We believe that our work will help to enable local industry partners to develop deep learning supported artificial intelligence solutions in the near future.

Keywords: Deep Learning, Artificial Intelligence, Course, Tensorflow, Kerras

1 Introduction

Artificial intelligence (AI) has made an astonishing development in recent years and surpassed human intelligence at a diverse set of difficult tasks ([1], [2] and [3]). Deep learning is the key technology that facilitates this progress. Deep learning uses a cascade of multiple layers of nonlinear processing units for feature extraction and transformation to solve supervised and unsupervised learning problems.

To prepare students for upcoming AI challenge, universities and universities of applied sciences started to offer AI and deep learning focused courses. These courses often focus on the mathematical foundations and technical details. An indication of such a theory-oriented course is examination in form of a verbal or written exam instead of project work. Theory-oriented deep learning courses may constitute several drawbacks. For instance, graduates may actually miss necessary hands-on programming, data handling and model optimization experiences to successfully apply deep learning in real-world industry applications. Furthermore, programming-oriented but mathematics-averse students may not have the mathematical background and might shy away from these courses and, thus, do not acquire deep learning capabilities at all.

Thus, the focus of this work is the design and implementation of a deep learning course with following preconditions and targets:

- Focus on practical exercises using cloud systems,
- As few mathematics as possible,
- Project work as examination,
- Non prior knowledge of machine learning,
- Focus on supervised learning applications, in particular image classification and natural language processing,
- Inverted classroom.

2 Methods and Materials

2.1 Computing Resources

Deep learning requires specific computing resources namely graphics processing unit (GPU). GPUs operate at lower frequencies but have many times the number of cores compared with traditional central processing unit (CPU). Thus, GPUs can process far more data per second than CPUs at lower purchase and operation costs. The University of Applied Sciences in Kaiserslautern has invested in such a GPU equiped computing resource (Table 1 for details) to facilitate deep learning projects and courses.

Table 1. Details of the deep learning server at Kaiserslautern University of Applied Sciences (price point about 20k€ in 2018).

Component	Description
CPU	28-Cores - 2x Intel Xeon E5-2680v4 (2.40-3.30GHz, 14-Cores)
RAM	256GB DDR4-RAM 2400MHz ECC Memory Reg. (16x32GB)
GPU	4x NVIDIA GTX 1080 Ti (11GB RAM 3.584-Cores)
SSD	2TB SSD SATA 6Gb/s
HDD	40TB HDD (Raid 5)
OS	Linux Ubuntu 16.04 LTS
POWER	4 * 1900W redundant power supply

In addition to on-premise computing resources, cloud resources were also employed. The Google Cloud Platform supplied education grants for both the teaching stuff (2 times 100 US$) and the students (25 times 50 US$). The Google Cloud Platform offers ready-to-use deep learning virtual machines with configurable computing resources and computing libraries that can be made available within a few minutes at nearly arbitrary scale. Student grants of 50 US$ provide about 50 to 100 hours of suitable computing resources.

2.2 Libraries and Toolset

The field of deep learning is a very active but immature research field, that besides computing power heavily relies on capable toolsets and software libraries. So-called deep learning libraries (e.g. Tensorflow ([4]), Kerras ([5]), Pytorch ([6]) and Caffe ([7]) among others) are of special importance as they facilitate the programming, training and inference of deep neural networks using very high (and comfortable) abstraction layers. For the purpose of an introductory lecture Tensorflow and Kerras were selected to be used. Tensorflow is the standard library in research and industry for deep learning applications. Furthermore, Tensorflow forms the basis of almost any other deep learning as backend. Kerras provides a more compact programming interface (compared with Tensorflow) that allows even more rapid programming but uses Tensorflow for all low-level computations.

Further, Numpy, SciPy ([8]) and Pandas ([9]) are required for process programming data preparation.

Jupyter notebooks ([10]) are another essential programming tool. These notebooks provide access to a web-based interactive code execution engine with markup language support for documentation and plotting capabilities.

2.3 Literature and Lecture Materials

Deep learning has a profound mathematical background that comprises elements of the fields of linear algebra, probability theory, numerical optimization and machine learning. Ian Goodfellow et al. provides an comprehensive text book on these preliminaries tailored to deep learning ([11]) and is freely available as HTML version.

The Massachusetts Institute of Technology (MIT) provides excellent deep learning online courses. The introductory course [12] contains seven lectures of the most relevant topics of deep learning and includes slides and videos of very high quality. Three of these lectures (detailed in Table 2) were used to implement the inverted classroom concept.

Table 2. Contents of the Massachusetts Institute of Technology lectures 6S191_L1, 6S191_L2 and 6S191_L3. These lectures are used for the inverted classroom concept [12].

Name	Content	Video Length
6S191_L1	The Perceptron Forward propagation Neural networks Loss optimization Backpropagation Regularization	45 minutes
6S191_L2	Sequence modells Recurrent neural networks LSTM networks	36 minutes
6S191_L3	Tasks of computer vision Convolutional neural network CNN architecture Beyond classification	41 minutes

To implement the practical parts of the course several programming exercises in the form of Jupyter notebooks were prepared [13]. Multiple public datasets were utilized including MNIST ([14]), CIFAR-10 ([15]) for convolutional neural networks (CNNs) and several freely available text books for recurrent neural networks (RNNs). These notebooks provide working examples of deep learning programming examples implementing the following procedure:

1. Data preparation with missing parts as programming exercises,
2. Presentation of a suboptimal neural network as the basis of programming exercises to analyze and optimize the network performance,
3. Presentation of a more suitable neural network as the basis of programming exercises to analyze the network performance.

2.4 Project work

Student projects for examination are based on reproducing, understanding and documenting results from existing and challenging deep learning application papers. A list of proposal papers for CNNs ([16],[17] ,[18],[19] ,[20]), RNNs ([21],[22],[23]) and Generative Adversarial Networks (GAN) ([24],[25]) were assembled.

3 Results

A deep learning course was designed to meet the targets and respect the preconditions as described in the introduction.

Table 3. Detailed outline of the full course. Each row describes a 180 minutes block divided into 45 minutes for theory upfront teaching and 135 minutes for hands-on practical programming exercises.

Nr.	Theory	Exercise & Homework
1	Motivation for deep learning Outline course Present student projects	Python, numpy, pandas Data preparation programming Homework: Finalize programming examples
2	Machine Learning Basics Regression and classification Test/train split Regularization	Machine Learning programming exercise Homework: Watch MIT lecture 6S191_L1
3	Recap of MIT lecture 6S191_L1 Model Optimization for DL Test/train split for DL	Dense networks Tensorflow Homework: Watch MIT lecture 6S191_L2
4	Recap of MIT lecture 6S191_L2 Introduction to Kerras	RNNs Kerras Homework: Watch MIT lecture 6S191_L3
5	Recap of MIT lecture 6S191_L3	Programming exercises CNNs Kerras
6	Teaser of further DL topics Project proposals	Project assignment
7-12	Status meetings with project teams	

In the first two lectures, the basic concepts of machine learning using the python machine learning technology stack are introduced and strengthened using multiple programming exercises. These two sessions are not specific to deep learning in particular and could also be placed at the beginning of a traditional machine learning course. Nevertheless, it is important to make the students familiar with the programming environment and to build a rudimentary understanding of the challenges of machine learning with respect to prediction, regularization and overtraining.

The next three lectures then use course materials (slides and videos) from the MIT introductory deep learning course (see Table 2 for details) to implement the inverted classroom concept. In these three sessions Neural Networks, RNNs and CNNs are introduced in this order, accompanied by extensive and suitable programming exercises. The MIT course material needs to be extended by theory chapters on model optimization techniques and procedures suitable for deep learning models. It is not expected that students fully understand all mathematical and technical details of the complex deep learning network architectures and inference procedures. The focus of the lectures needs to be on the programming exercises. Students utilize cloud systems to perform programming exercises. Cloud systems are required to enable more than a hand full of students to work in parallel on deep learning challenges.

With the sixth session the course enters its project phase. Students in groups of two to four are assigned to a deep learning project. The preparation of practically relevant and interesting student projects with a suitable level of complexity is a major challenge that was addressed. Project progress is monitored and project support is provided with bi-weekly status meetings.

4 Conclusions

A concept for a deep learning course was presented that addresses current industry needs and is suitable for a computer science master program of a university of applied sciences.

The authors believe that this work will help to enable local industry partners to develop deep learning supported AI solutions in the near future. Finally, the work presented here might motivate and help other education facilities to offer similar programs to support European AI initiatives at a wider scale. Feel free to contact the authors for friendly exchange of experience, advice or course materials.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR **abs/1502.01852** (2015)
2. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nature **529**(7587) (January 2016) 484–489
3. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
4. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from tensorflow.org.
5. Chollet, F., et al.: Keras. https://github.com/fchollet/keras (2015)
6. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)
7. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22Nd ACM International Conference on Multimedia. MM '14, New York, NY, USA, ACM (2014) 675–678
8. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001–) [Online; accessed ¡today¿].
9. McKinney, W.: Data structures for statistical computing in python. In van der Walt, S., Millman, J., eds.: Proceedings of the 9th Python in Science Conference. (2010) 51 – 56
10. Pérez, F., Granger, B.E.: IPython: a system for interactive scientific computing. Computing in Science and Engineering **9**(3) (May 2007) 21–29
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016) http://www.deeplearningbook.org.
12. Amini, A., Soleimany, A., et al.: 6.s191: Introduction to deep learning. http://introtodeeplearning.com (2018)
13. Müller, R., Trefz, F.: Projektbericht zum projekt deep learning (2019)

14. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. (1998) 2278–2324
15. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12, USA, Curran Associates Inc. (2012) 1097–1105
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
18. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4) (April 2017) 664–676
19. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. CoRR **abs/1603.08511** (2016)
20. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS'13, USA, Curran Associates Inc. (2013) 2553–2561
21. Deshpande, A.: How i used deep learning to train a chatbot to talk like me. (2017)
22. Sutskever, I., Martens, J., E. Hinton, G.: Generating text with recurrent neural networks. (01 2011) 1017–1024
23. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR **abs/1406.1078** (2014)
24. Dahl, R., Norouzi, M., Shlens, J.: Pixel recursive super resolution. CoRR **abs/1702.00783** (2017)
25. Denton, E.L., Chintala, S., Szlam, A., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. CoRR **abs/1506.05751** (2015)