EXPLORING THE BORDER BETWEEN DEEP AND SHALLOW LEARNING

LEREN EN BESLISSEN PROJECT 2020

Stijn Verdenius (BrainCreators / UvA)

December 2019

ABSTRACT
"If the only tool you have is a hammer, everything is treated as if it were a nail". The past decade has seen a hype on neural networks like never before. Although there is a good reason for this, as they are very powerful, sometimes deep learning is not the answer. In this project, students will explore what kind of tasks and data-patterns make the one or the other superior, and try to find where the tipping-point is.

Contents

1 Description

There has been a lot of hype for the application of deep learning with the successes it has achieved in Computer Vision (CV) and Natural Language Processing (NLP) since the revolution of AlexNet (Krizhevsky, Sutskever, & Hinton, 2012). However, this doesn't mean that every problem necessarily needs to be solved by neural networks. For some problems, it has long been known that shallow learning works better, yet for others that deep learning does. Having said that, maybe in some practical occasions we would want to employ the other type of learning to benefit from some of the convenient properties it has. Think for example, of properties like 'automatic feature extraction' (deep learning) or 'interpretability' (shallow learning). This project will be about exploring the border between the two, to maybe leverage problem and data properties to make better design decisions in practical ML applications.

For a couple of tasks and datasets, students will compare the performance of deep and shallow learning. Why is a certain model better at a certain task? Where is the limit? What are the properties that define that? Can we control these properties to our advantage? Some things to start off with looking at are; the performance-gain per added model-parameter, feature extraction efforts, interpretability, scale-ability, complexity and data-hunger. However, students are encouraged to discover their own distinctions as well.

As for now, we would recommend between 2 and 4 of the following experiments, which will be decided upon in consultation with the students at the start of the project[1]:

- MNIST digit classification (see also). We will compare Logistic Regression (Bishop, 2006) vs. the LeNet-5 model (LeCun, Bottou, Bengio, Haffner, et al., 1998) or a simple fully-connected neural network (FC)(Bishop, 2006).
- Stanford Sentiment Classification: LSTM (Hochreiter & Schmidhuber, 1997) vs. simple N-gram Naive Bayes (see blogpost)
- Property Inspection Prediction: XGBoost (see implementation and explanation) vs FC neural nets (Bishop, 2006)
- Language Model text generation on some student-chosen corpus with Perplexity scores: LSTM (Hochreiter & Schmidhuber, 1997) vs. N-gram based Naive Bayes next-word prediction
- CiteULike-a recommendation like in the experiment in Dacrema, Cremonesi, and Jannach (2019), with HR@5 and NDCG@5 scores: KNN (Bishop, 2006) vs. CMN (see implementation here)

Please note, that students are not expected to have any prior knowledge on deep learning concepts such as backpropagation and different layer or model designs. Any prior knowledge that is deemed relevant, will be taught to students at the start of the project. To an extent, deep learning modules will be provided by the company, and code online may freely be used to their own liking. If any further difficulties may arise, then students will be guided with those too.

[1] If students have an idea of their own for task-dataset, this can also be discussed.

2 Goal

Give a overview of properties of a problem/data-set that makes it suitable for deep learning and not shallow learning and visa versa. Moreover, identify the tipping points for these properties and define some conditions we could test on, in the form of some code or even just analytically. The project is mainly about the analysis that students provide.

3 Learning Goals

- Not all problems need to be solved by deep learning.
- However, sometimes they should
- Learning to spot the tell-signs for this distinction
- Doing analytical ML-research
- Practical experience with ML models
- Introduction into some fields of application (e.g. CV, NLP & IR)

4 Tasks

There are three steps the students will go through for this project. Not all are required, but we obviously hope that students will be interested and try to take on as many as they can. Task-details will be further specified in collaboration with students at the start of the project. The three tasks are:

1. **Basic:** Students will implement and run the aforementioned experimental designs. They will do this with provided deep-learning models in PyTorch for the deep side and with own implementation or code found online (e.g. sklearn) for the shallow side. Subsequently, they will analyze why deep or shallow wins, on the hand of the tasks, model and dataset properties. Students will get guidance with the selection of properties to look at, but are encouraged to come up with new things as well.

2. **Default:** Can the students identify which properties make the outcome of experiments swap from being better solved by deep and shallow. Properties may be found in model hyperparameters as well as input-features, subsets of training-data or statistical patterns in data. It will probably be done by means of ablation study, clever training-data selection or some form of statistical analysis. Students will get guidance with these concepts.

3. **Advanced:** Can we generate a simple dataset (e.g. maybe pictures of circles and squares instead of MNIST for the CV-task) with two of the mentioned tasks; one winner for deep and one winner for shallow, that if we turn some buttons for some data properties - which the students identified before - it will also swap which algorithm (deep vs. shallow) is better at said task. I.e. can we actually control the tipping-point between deep- and shallow strategies for controlled generated data? We will provide data-generation tools (GANs, VAEs etc) where needed.

5 Additional Reading Material

Deep vs. Shallow Learning Blogposts:

- Discussion favouring shallow learning: [2] `https://www.quora.com/Is-deep-learning-always-better-than-traditional-machine-learning`
- Article explaining the pros of deep learning: [2] `https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063`

Introductions To Some Concepts:

- Blogpost to introduce deep learning: [2] `https://towardsdatascience.com/deep-learning-101-for-dummies-like-me-a53e3caf31b1`
- Thorough introduction to deep learning: [2] `https://www.3blue1brown.com/neural-networks`
- Small introduction to reccomendation systems (IR subfield): [3] `https://medium.com/@madasamy/introduction-to-recommendation-systems-and-how-to-design-recommendation-system-that-resembling-the-9ac167e30e95`
- Small introduction into transfer-learning (used to migitate some problems of deep learning): [3] `https://medium.com/kansas-city-machine-learning-artificial-intelligen/an-introduction-to-transfer-learning-in-machine-learning-7efd104b6026`
- Introduction into recurrent networks and LSTM cells (used for NLP tasks): [3] `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`
- Introduction into Convolutional layers (used for CV tasks): [3] `https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7`

Algorithmic Reference Material:

- Machine learning book by Bishop (2006) for theoretical foundation of ML methods. [3]
- Website of pytorch (`https://pytorch.org/`) for deep learning implementation [2]

Some Papers:

- Paper on deep vs. shallow in Recommendation by Dacrema et al. (2019) [3]
- Paper on performance of deep vs shallow in extreme conditions by Banerjee, Bhattacharjee, and Das (2017) [3]
- Paper on deep vs shallow learning in (medical) imaging by Chauhan et al. (2019) [3]
- Paper on deep vs deeper learning by Ba and Caruana (2014) [3]

[2] Recommended reading
[3] Extra reading

References

Ba, J., & Caruana, R. (2014). Do deep nets really need to be deep? In *Advances in neural information processing systems* (pp. 2654–2662).
4

Banerjee, S., Bhattacharjee, P., & Das, S. (2017). Performance of deep learning algorithms vs. shallow models, in extreme conditions-some empirical studies. In *International conference on pattern recognition and machine intelligence* (pp. 565–574).
4

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer Science+ Business Media.
2, 4

Chauhan, S., Vig, L., De Grazia, M. D. F., Corbetta, M., Ahmad, S., & Zorzi, M. (2019). A comparison of shallow and deep learning methods for predicting cognitive performance of stroke patients from mri lesion images. *Frontiers in neuroinformatics*, *13*.
4

Dacrema, M. F., Cremonesi, P., & Jannach, D. (2019). Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th acm conference on recommender systems* (pp. 101–109).
2, 4

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.
2

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).
2

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.
2