University of Groningen

Research Internship

On-line Learning under Concept Drift

Author:
Pieter Jan Eilers
(s2381575)

Supervisors:
prof. dr. Michael Biehl
Michiel Straat, Msc

July 31, 2020

Abstract

Using a modeling framework for the purpose of investigating on-line learning processes in non-stationary environments, we conduct experiments for a number of different situations. We consider the learning of a regression scheme in layered neural networks using sigmoidal and ReLU activation. In all situations, the target, i.e. the regression scheme, changes continuously while the system is trained from a stream of input data. We run Monte Carlo simulations in Student-Teacher scenarios equal number of student and teacher units, $K = M$. We extend this to the overlearnable case, where $K > M$. We include weight decay as a from of explicit forgetting and study its effects with regards to drift.

1 Introduction

Feedforward neural networks are heavily used tools for the purpose of classification and regression. Shallow networks of only one hidden layer are already sufficient to represent non-trivial scalar function of N-dimensional variables [1]. However, the convergence of the training of such networks can be very slow due to the occurrence of so-called *plateau states*. In terms of on-line learning in a student-teacher scenario, the student can get stuck in an unspecialized local optimum before rapidly specializing to the hidden teacher unit.

In this project, we consider two choices for activation functions in the hidden units. Conventionally, sigmoidal activation functions have been used, however, recently ReLU activation has gained popularity, mostly due to improved empirical performance, for example in [2]. Theoretical advantages have also been shown in [3].

Generally, a machine learning process can be separated into two stages, the training phase and the test phase [4]. In the training phase, example data is presented and analyzed, information is extracted and a hypothesis is parameterized in terms of a classifier, or a regression scheme. In the following stage, the working phase, this hypothesis can be tested with novel data. This process implicitly assumes that the training set does not change, i.e. the statistic properties of the data and the actual target task remains the same after training. However, this is not always the case in machine learning tasks and it is not a plausible assumption with regards to the way learning happens in humans and other biological processes. In such situations, the learning system must be able to detect and track *concept drift*, i.e. forget irrelevant, older information while continuously adapting to more recent inputs. This process is also known as *continual learning* or *lifelong learning*. The theoretical properties and statistical mechanics of concept drift have been studied before [5, 6]. In this project, the focus lies on practical simulations of student-teacher scenarios for the purpose of learning a regression scheme in varying situations.

2 Models and Methods

In the following sections, we present a student-teacher scenario for the learning of a regression scheme [7] with shallow feedforward neural networks. We explain and compare two types of hidden activation units, sigmoidal transfer functions and the popular rectified linear unit, or ReLU. This project considers gradient-based training of a Soft Committee Machine in the presence of real concept drift with the inclusion of weight decay as a mechanism of explicit forgetting.

2.1 Soft Committee Machines

A feedforward neural network with a non-linear activation function, a single hidden layer and a linear output unit is known as a Soft Committee Machine (SCM). Its structure resembles that of committee machine with binary threshold hidden units, where the network's response is given by their majority vote. The output of a SCM with K hidden units and fixed hidden-to-output weights can be defined as

$$y(\boldsymbol{\xi}) = \sum_{k=1}^{K} g(\mathbf{w}_k \cdot \boldsymbol{\xi}) \tag{1}$$

where $\mathbf{w}_k$ denotes the weight vector connecting this input data to the k-th hidden unit. A non-linear activation function $g(x)$ defines the hidden unit states and the final output is given as their sum. For the sigmoidal case, this equates to

$$g(x) = \mathrm{erf}(x/\sqrt{2}),\ g'(x) = \sqrt{\frac{2}{\pi}} e^{-x^2/2}. \tag{2}$$

The ReLU activation function is defined as

$$g(x) = x\theta(x),\ g'(x) = \theta(x), \tag{3}$$

where $\theta(x)$ is the step function, defined as

$$\theta(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

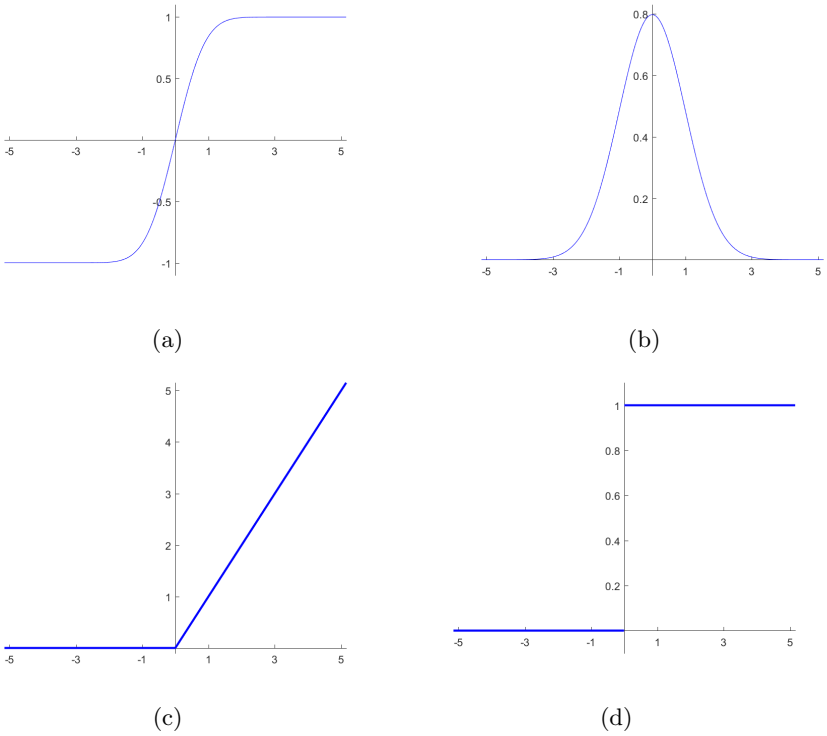


Figure 1: (a) Sigmoidal activation using the erf function, (b) The first derivative of erf(x), (c) ReLU activation: $x\theta(x)$, (d) The first derivative of the ReLU activation function: $\theta(x)$

Both activation functions have their advantages and disadvantages [8] and their characteristics with the inclusion of drift can vary, as shown in section 3. In Figure 1 we can see the plots of the activation functions and its derivatives. We observe that the first derivative of the erf function is Gaussian in nature, while the ReLU derivative is 0 for $x < 0$ and 1 for $x > 0$. One property to account for is the discontinuity of ReLU$'(0)$, in practice, one can choose either 0 or 1 in this situation.

2.1.1 On-Line Learning

The training of a neural network with real-valued output $y(\boldsymbol{\xi})$ based on examples $\{\boldsymbol{\xi}^\mu \in \mathbb{R}^N, \tau^\mu \in \mathbb{R}\}$ is generally guided by the quadratic deviation of the network output from the rule output. This deviation serves as a cost function which evaluates the network performance with respect to a single example:

$$e^\mu(\{\mathbf{w}_k\}_{k=1}^K) = \frac{1}{2}(y^\mu - \tau^\mu)^2 \text{ with } y^\mu = y(\boldsymbol{\xi}^\mu). \tag{4}$$

In on-line gradient descent, updates of the weight vectors are based on the presentation of a single example at timestep μ

$$\mathbf{w}_k^\mu = \mathbf{w}_k^{\mu-1} + \eta/N \Delta \mathbf{w}_k^\mu \text{ with } \Delta \mathbf{w}_k^\mu = -\frac{\partial e^\mu}{\partial \mathbf{w}_k} \tag{5}$$

where η represents the learning rate. For the SCM architecture specified in Eq. (1), $\partial y^\mu / \partial \mathbf{w}_k = g'(h_k^\mu)\xi^\mu$ and we obtain

$$\Delta \mathbf{w}_k^\mu = -\left(\sum_{i=1}^{K} g(h_i^\mu) - \tau^\mu \right) g'(h_k^\mu) \boldsymbol{\xi}^\mu \tag{6}$$

with the inner products $h_i^\mu = \mathbf{w}_i^{\mu-1} \cdot \boldsymbol{\xi}^\mu$ of the current weight vectors with the next example input in the stream. This change of weight vectors is proportional to $\boldsymbol{\xi}^\mu$ and can be interpreted as a form of Hebbian learning [4].

2.2 Student-Teacher Scenario

In order to define and model meaningful learning situations we resort to the consideration of student-teacher scenarios. We assume that the target can be defined in terms of an SCM with a number M of hidden units and a specific set of weights $\{\mathbf{B}_m \in \mathbb{R}^N\}_{m=1}^M$:

$$\tau(\boldsymbol{\xi}) = \sum_{m=1}^{M} g(\mathbf{B}_m \cdot \boldsymbol{\xi}) \text{ and } \tau^\mu = \tau(\boldsymbol{\xi}^\mu) = \sum_{m=1}^{M} g(b_m^\mu) \tag{7}$$

with $b_m^\mu = \mathbf{B}_m \cdot \boldsymbol{\xi}^\mu$. There are three different student-teacher scenarios with regards to hidden units. $M > K$ hidden units, an unlearnable target, where the students can not perfectly align with the teacher. On the contrary, $K > M$ would correspond to an overlearnable target, also known as overfitting. In practice, one usually does not know the complexity of the task, making this an interesting case to study. The last and most studied scenario has $K = M$, where the two architectures match and the student has to ability to fully represent the rule, without any redundancies. In this project, the focus lies on scenarios with $K = M$ and $K > M$.

2.3 Order parameters

The many degrees of freedom, i.e. the components of the adaptive vectors, can be characterized in terms of only very few quantities. The definition of these so-called order parameters follows naturally from the mathematical structure of the model. The order parameters quantify the similarity of student weight vectors with other student weight vectors and the similarity between student and teacher weight vectors. After presentation of a number μ of examples, the order parameters are

$$R_{im} = \mathbf{w}_i \cdot B_m, \quad Q_{ik} = \mathbf{w}_i \cdot \mathbf{w}_k \quad m = 1, \ldots, M \quad i, k = 1, \ldots, K. \tag{8}$$

2.4 Generalization Error

After training, the success of learning is quantified in terms of the generalization error ϵ_g, which is also given as a function of the order parameters. The generalization error can be defined as the average of the quadratic deviation between student and teacher output over the isotropic density [5]. For the sigmoidal case, this equates to

$$\begin{aligned}\epsilon_g =& \frac{1}{\pi}\Bigg[\sum_{i,j=1}^{K} sin^{-1}\left(\frac{Q_{ij}}{\sqrt{1+Q_{ii}}\sqrt{1+Q_{jj}}}\right) \\ & - 2\sum_{i=1}^{K}\sum_{m=1}^{M} sin^{-1}\left(\frac{R_{im}}{\sqrt{1+Q_{ii}}\sqrt{2}}\right)\Bigg] \\ & + \frac{M}{6}\end{aligned} \tag{9}$$

For the ReLU case,

$$\begin{aligned}\epsilon_g =& \frac{1}{2}\Bigg[\sum_{i,j=1}^{K}\left(\frac{Q_{ij}}{4} + \frac{\sqrt{Q_{ii}Q_{jj}-Q_{ij}^2}}{2\pi} + \frac{Q_{ij}\sin^{-1}\left(\frac{Q_{ij}}{\sqrt{Q_{ii}Q_{jj}}}\right)}{2\pi}\right) \\ & - 2\sum_{i=1}^{K}\sum_{m=1}^{M}\left(\frac{R_{im}}{4} + \frac{\sqrt{Q_{ii}-R_{im}^2}}{2\pi} + \frac{R_{im}sin^{-1}\left(\frac{R_{im}}{\sqrt{Q_{ii}}}\right)}{2\pi}\right) \\ & + \frac{M}{2} + \frac{(M-1)M}{2\pi}\Bigg].\end{aligned} \tag{10}$$

Both equations are for orthonormal teacher vectors, where $\mathbf{B}_m \cdot \mathbf{B}_m = 1$ and $\mathbf{B}_m \cdot \mathbf{B}_n = 0$ for $m \neq n$. Extensions to general teacher vectors exist [9].

2.5 Drift

The models, as explained in the previous sections, concern learning in a stationary network, where the characteristic vectors $\mathbf{B}_m$ do not change during the course of the training. We would like to extend this to non-stationary environments, which can be divided into two categories, virtual drift and real drift. Virtual drift affects the statistical properties of the observed example data, while the actual target function remains unchanged. In this project we focus on the other category, *real drift*. Here, the actual target changes, i.e. the characteristic vectors $\mathbf{B}_m$ displaces over time. A variety of time-dependencies could be considered in the model. Important to note is that, in practice, real drift processes are often accompanied by virtual drift, see [10] for an overview. We restrict ourselves to the analysis of diffusion-like random displacements of vectors $\mathbf{B}_m(\mu)$

at each time step. Upon presentation of example μ,we assume that random vectors B_m are generated which satisfy the conditions

$$\begin{aligned} &\mathbf{B}_m(\mu) \cdot \mathbf{B}_m(\mu-1) = (1-\delta/N) \\ &\mathbf{B}_m(\mu) \cdot \mathbf{B}_n(\mu-1) = 0, \text{ for } m \neq n = 0 \text{ and } |\mathbf{B}_m(\mu)|^2 = 1 \end{aligned} \tag{11}$$

with $m, n = 1..M$. Here δ quantifies the strength of the drift process. The orthonormality of the teacher vectors is preserved in the drift.

2.6 Weight Decay

We include weight decay as to enforce explicit forgetting and to potentially improve the performance of the systems in the presence of real concept drift. We consider the multiplication of all adaptive vectors by a factor $(1-\gamma/N)$ before the generic learning step given by Eq. (5),

$$\mathbf{w}_i^{\mu} = (1-\gamma/N)\mathbf{w}_i^{\mu-1} + \eta/N\Delta\mathbf{w}_i^{\mu}. \tag{12}$$

The multiplications with $(1-\gamma/N)$ accumulate in the course of training, thereby enforcing an increased influence of the most recent training data as compared to earlier examples.

2.7 Methodology

For the purpose of this project, a framework has been developed in which we can run Monte Carlo simulations. Random input vectors $\boldsymbol{\xi} \in \mathbb{R}$ are generated and both student and teacher weights with preset overlap. Here we use

$$R_{im} = 0, \quad Q_{ii} = 0.5, \quad Q_{ik} = 0.49, \text{ for } i \neq k, \tag{13}$$

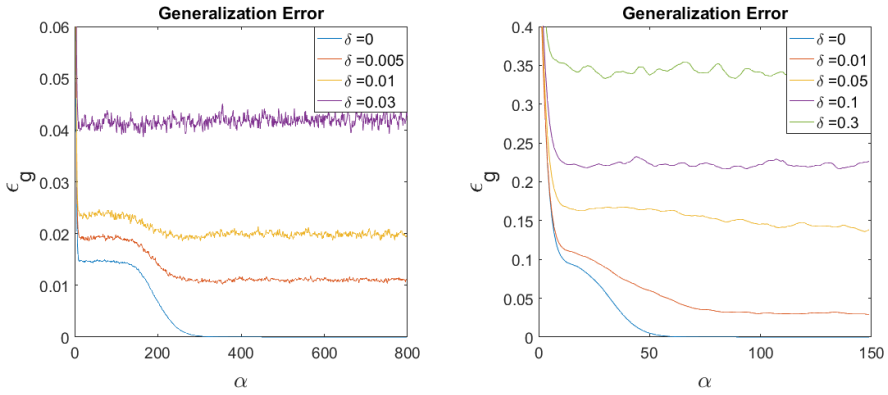
where $i, k = 1..K$ and $m = 1..M$. This means that the students have no prior knowledge of the rule and a large amount of overlap between themselves, resulting in longer plateau states. We use a generalized version of the Gram-Schmidt method to initialize the overlap. Different parameters can be tweaked to compare different initializations, the drift strength δ, the learning rate, γ and the amount of student vectors and hidden teacher units. Furthermore, the total amount of timesteps μ can be adapted. The resulting graphs generally have $\alpha = \mu/N$ along the x-axis, this scaling corresponds to the assumption that the number of examples require for successful training is proportional to the number of degrees of freedom in the system.

3 Results

Here we present the results obtained from the Monte Carlo simulations of online learning with drift. We distinguish between two important cases, firstly, $K = M$, the exact case with equal student and teacher vector. We also include the overrealizable case, $K > M$, and its relation with concept drift. In all of the following experiments, we use system size $N = 500$.

3.1 $K = M$

Firstly, we look at the matching case, where $K = M = 2$ and the two architectures match. In Figure 2 we can see the generalization error ϵ_g over time for both sigmoidal and ReLU activation. The results are for simulations run with $\eta = 0.5$, $\alpha = 800$ for sigmoidal activation and $\alpha = 150$ for ReLU activation. These results are then averaged over 5 runs. We observe that, as the drift strength increases, the final generalization error also increases.



(a) (b)

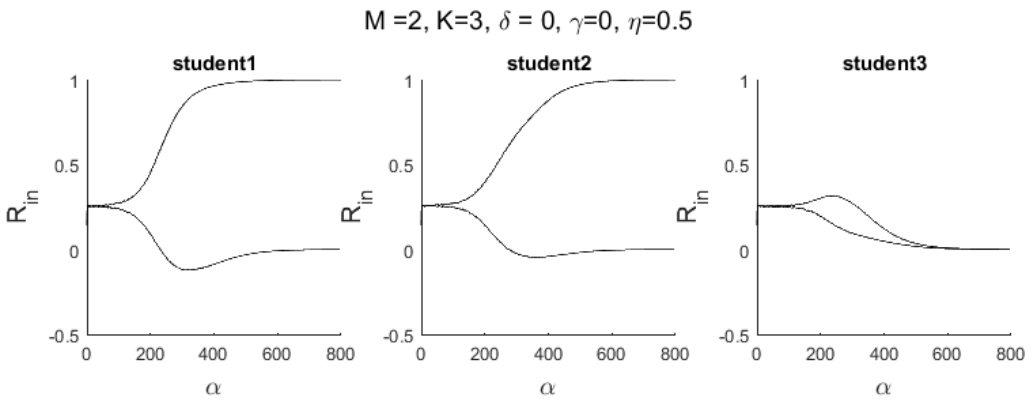
Figure 2: (a) Generalization error for sigmoidal activation, for different δ's, (b) ReLU activation

3.2 $K > M$

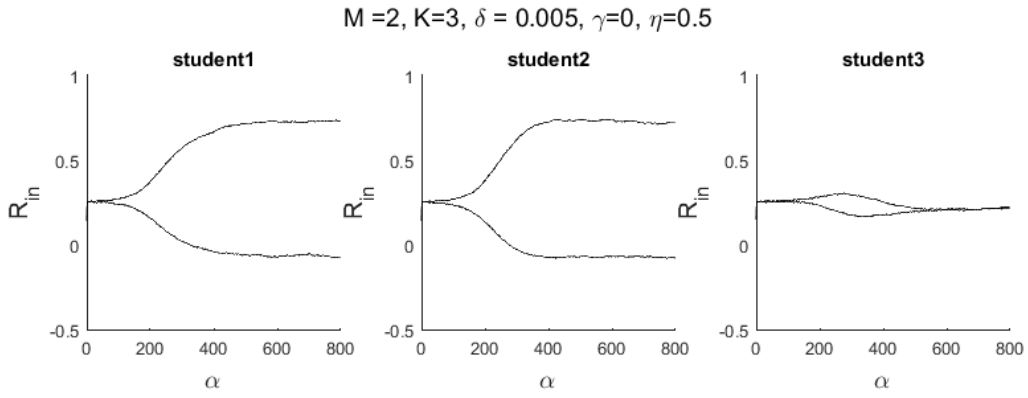
In the overrealizable scenario, we can differentiate between two situations. The case in which K is not a multiple of M, for example, $K = 3$ and $M = 2$. The second situation has K as a multiple of M, $K = k * M$, $k \in \mathbb{Z}^+$. These two cases can have interesting variations between them, especially in terms of the order parameters R_{in} and Q_{ik}.

3.2.1 K Not a Multiple of M

In the sigmoidal case, with $K = 3$ and $M = 2$, regardless of drift strength, two students will specialize to 2 teachers, while one student will get phased out, specializing to no teachers. This is shown in Figure 3, which displays the plots of the order parameters R_{im}. The plot shows the results for simulations with $\eta = 0.5$ and $\delta = 0.005$ and averaged over 5 runs.



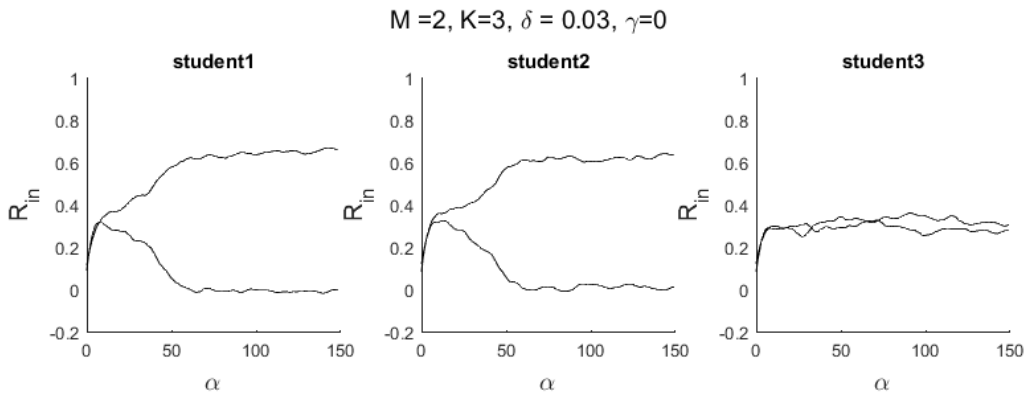
(a)



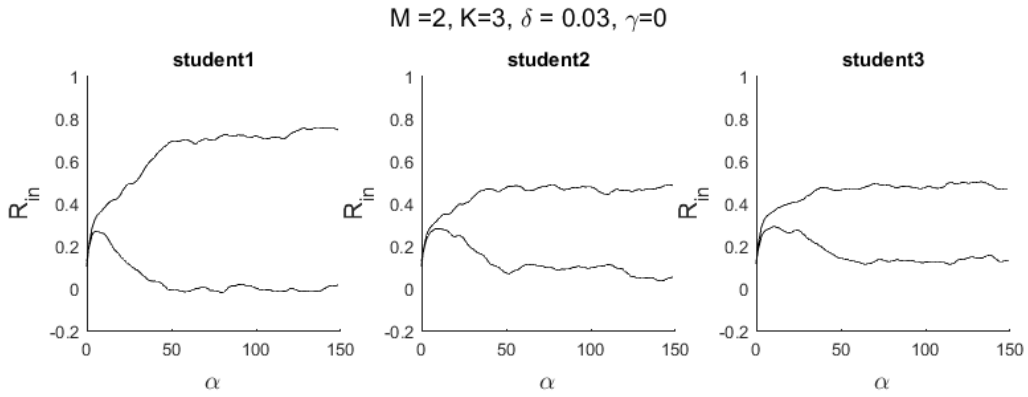
(b)

Figure 3: (a) Order parameters R_{in} for sigmoidal activation and $K > M$, without drift, (b) A drift is introduced with $\delta = 0.005$.

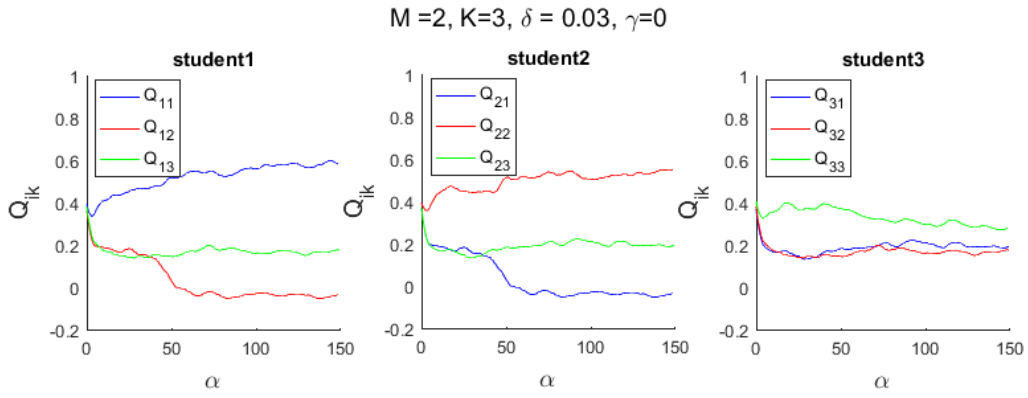
In the ReLU case, two different situations occur in the presence of drift. The first situation being similar to the sigmoidal case, where 2 students will specialize to 2 teachers, and the remaining students get phased out. In the other situation, 1 student will fully specialize to 1 teacher, while the other 2 students will share the specialization to the remaining teacher, as shown in Figure 4b and 4a for $\eta = 0.5$ and $\delta = 0.03$. Below these figures we can see the order parameters Q_{ik}, showing overlap between students, here the blue line indicates the overlap with the first student, red the second and green the third. Because of these 2 differing cases that can occur, these simulations have not been averaged over multiple runs.



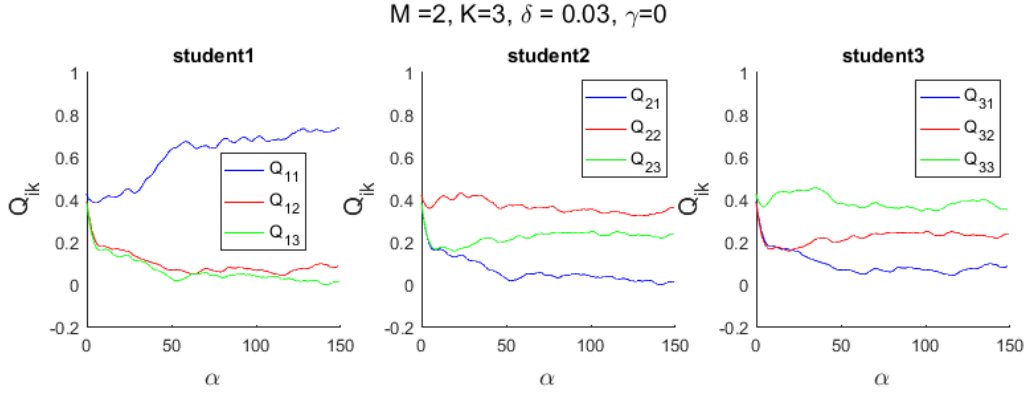
(a)



(b)



(c)



(d)

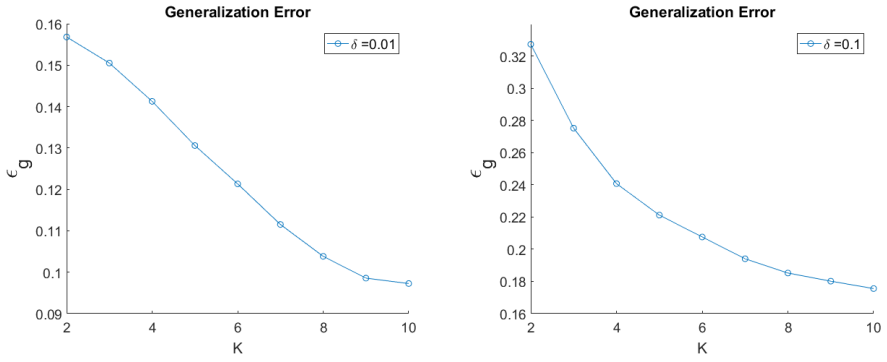
Figure 4: (a) Order parameters R_{in} where 1 student is unspecialized, (b) Order parameters R_{in} where 2 students share specialization, (c) Order parameters Q_{ik} where 1 student is unspecialized, (d) Order parameters Q_{ik} where 2 students share specialization

3.2.2 K is a Multiple of M

When $K = kM$, $k \in \mathbb{Z}^+$, the order parameters mostly simplify. With sigmoidal activation, the students that are phased out can still have overlap with the teacher, where 1 student has positive overlap and the other has a negative overlap, as displayed in Figure 5a with the red line for $\eta = 0.5$, $\delta = 0.005$ and $\alpha = 800$. For ReLU activation, the situation is essentially identical for all drift strengths that allow specialization, where 2 students will share 1 teacher unit, as displayed in 5b, where $\eta = 0.5$, $\delta = 0.05$ and $\alpha = 150$.

3.2.3 Effects of Higher K

As the complexity of the rule is often unknown in practice, situations where $K > M$ can occur. Furthermore, with the inclusion of drift, it might even serve more useful to have an overlearnable target. In Figure 6 we can see the final generalization error as a function of K. In this simulation, we use a smaller learning rate, $\eta = 0.1$. We use $\alpha = 1200$ for sigmoidal activation and $alpha = 500$ for ReLU.

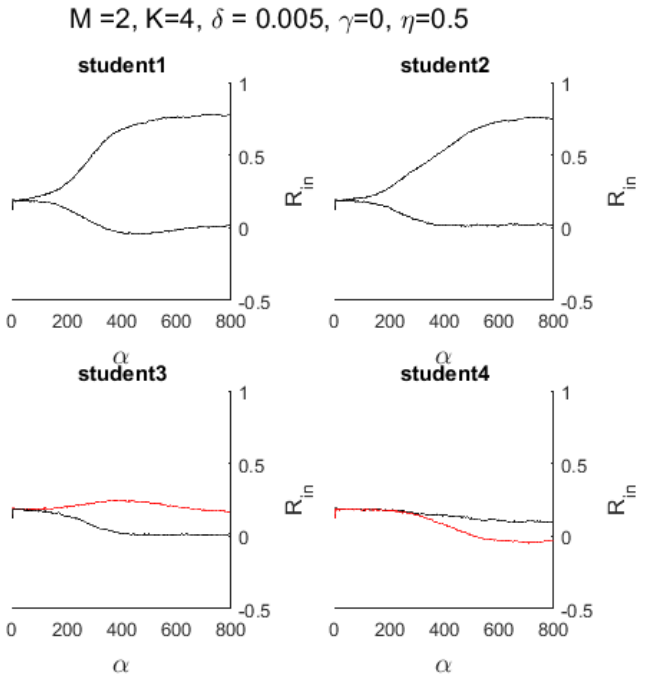


(a) (b)

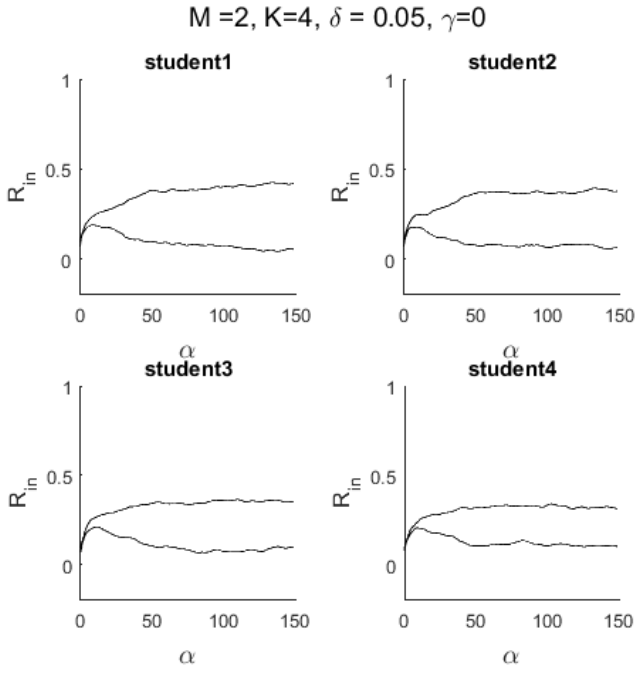
Figure 6: (a) Final generalization error versus K for sigmoidal, (b) ReLU

3.3 Optimal Learning Rate

Figure 7 shows the optimal learning rate as a function of the generalization error, for 2 situations. In the first one, $K = M = 2$, and in the second one, $K > M$, with $K = 4$ and $M = 2$. In both cases, $\alpha = 800$ for sigmoidal and $\alpha = 300$ for ReLU. The final generalization error is averaged of the last 75000 timesteps in which the students have achieved optimal overlap with the teachers.



(a)



(b)

Figure 5: (a) Order parameters R_{in} for $K = 4$, $M = 2$, the red line shows the non-specialized students can cancel each other out with positive and negative overlaps, sigmoidal. (b) Order parameters R_{in}, $K = 4$, $M = 2$ for ReLU

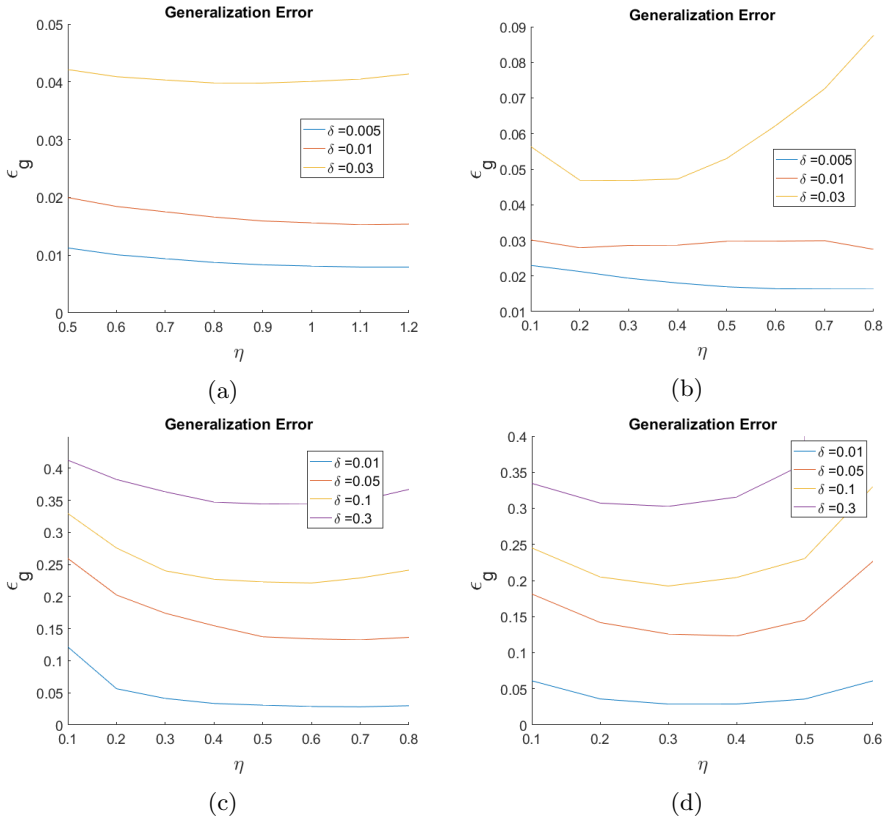


Figure 7: (a) Generalization error versus learning rate, for different δs. Sigmoidal activation $K = M = 2$, (b) Sigmoidal activation, $K = 4, M = 2$, (c) ReLU activation, $K = M = 2$, (d) ReLU activation, $K = 4, M = 2$.

3.4 Effect of Weight Decay

Figure 8 shows the effect if weight decay on the final generalization error for both ReLU and sigmoidal activation. Here, $K = 2, M = 2$, $\eta = 0.5$, $\alpha = 800$ for sigmoidal and $\alpha = 300$ for ReLU. The final generalization error is again averaged of the last 75000 timesteps in which the students have achieved optimal overlap with the teachers.

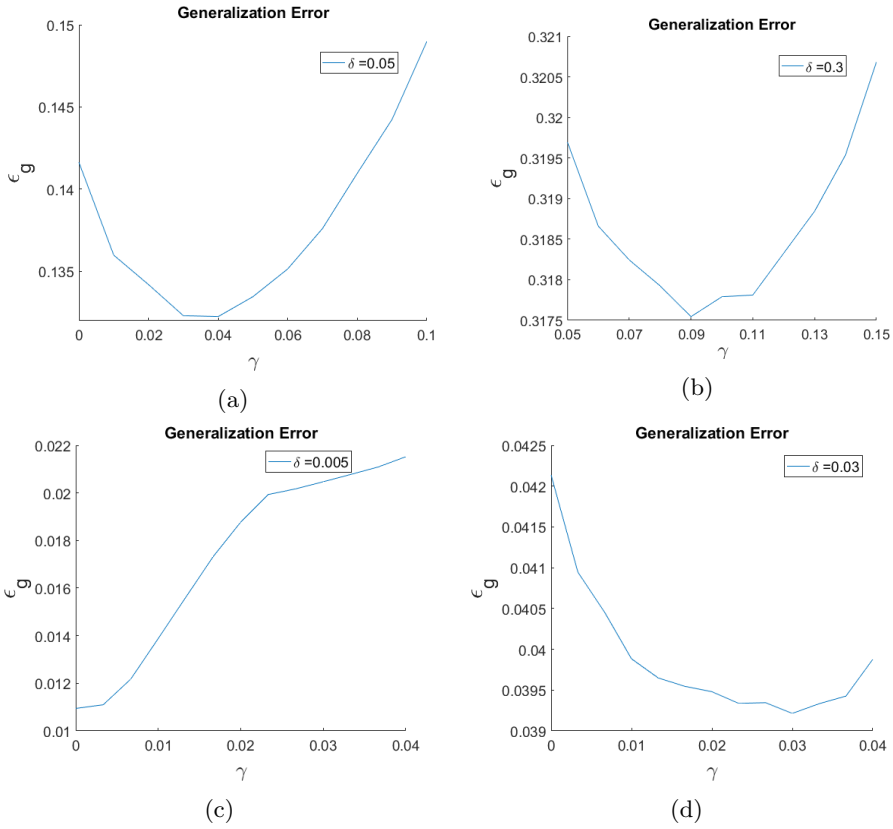


Figure 8: (a) Generalization error versus weight decay, ReLU, $\delta = 0.05$, (b) ReLU, $\delta = 0.3$ (c) Sigmoidal, $\delta = 0.005$, (d) Sigmoidal, $\delta = 0.03$

4 Discussion

Starting with the $K = M$ situation, from Figure 2 we can observe that ReLU activation is better able to handle real drift. As with sigmoidal activation, for $\delta > 0.03$, the SCM remains unspecialized and the achievable generalization ability is quite poor. In the ReLU case, for higher δ's, it might look like the same is happening. However, after a rapid decrease, a short plateau is reached and this symmetry is quickly broken before the system reaches its final generalized state. This is more clear in the paper by Michiel Straat et al. [5], where a smaller learning rate is used.

Looking at overlearnable scenario, where we have more students than teacher vectors, non-trivial situations do occur. There seems to be a notable difference between sigmoidal and ReLU when $K > M$ and $K \neq kM$, $k \in \mathbb{Z}^+$. Sigmoidal activation will have 2 students specializing to 2 teachers, for all drift strengths

that still allow specialization. Whereas with ReLU activation, when we include drift, we can distinguish between 2 cases. The first one being identical to the sigmoidal case, and the second case, where 2 students can share specialization of 1 hidden teacher unit. From the order parameters Q_{ik} in Figure 4d we can see that student 1 has no overlap with student 2, while both student 1 and 2 have a small overlap with student 3 which can likely be attributed to the drift. In the other situation, showcased in Figure 4c we can see that student 1 has no overlap both student 2 and 3, whereas student 2 and 3 have overlap between themselves, this is logical, since student 1 specializes to 1 hidden teacher unit and student 2 share specialization to the other teacher unit. When K is a multiple of M, the situation simplifies for ReLU. Figure 5b shows that for $K = 4$ and $M = 2$, 2 students will share specialization with one teacher and the other 2 students will share specialization with the remaining teacher unit. This is the case for all drift strengths that still allow specialization.

As shown in Figure 6, it might serve useful to have an overlearnable target. In practice, one rarely knows the complexity of the rule. So when one suspects there is drift in the learning process, increasing K to improve generalization can be considered. A consequence of this is the increased risk of overfitting

In Figure 7, we can see that, for $K = 3$ and $M = 2$, the optimal learning rate with significant drift is around 0,5, for both ReLU and sigmoidal. As the number of hidden units K increases, lower learning rates are more ideal. Furthermore, the threshold for divergence will decrease with increasing K, a learning rate that is too high will not facilitate any progress and the student weight vectors will actually diverge from the rule.

Looking at Figure 8, we can observe that larger amounts of drift benefit from a higher value of γ. As ReLU is generally more able to deal with larger drift, it also benefits more from a higher value of γ in those situations. This is logical, if a system is highly non-stationary, it has to be more able to forget old data. Since sigmoidal activation does not handle drifts of $\delta > 0.01$ in a desirable manner, weight decay has little to no effect. For low amounts of drift, any weight decay actually increases the final generalization error.

5 Outlook

This project focused mostly on the effect of real drift with regards to on-line learning of a regression scheme using two different activation functions. Especially, the differences between the matching and overlearnable situations, and the possible advantages and disadvantages between them. There is a variety of other relevant possible projects in this field. Here, we present an short summary of interesting future projects.

- The comparison of simulations with the theoretical description of the learning dynamics [5], where $N \to \infty$.
- The effect of weight decay and drift on the actual plateau lengths and the generalization error ϵ_p in these plateau states.
- Deterministic concept drifts, similar to the processes in the context of perceptron training[11–13]. This way, learning from an *adversary* can be modelled, where the modification of the target depends explicitly on the actual student configuration.
- As deep learning [14] has been an interesting topic of research in recent years, the extension to a deeper SCM architecture, with more than one hidden layer is an important forthcoming study.
- A more practical approach can be researched, with realistic data streams, to infer whether the results in this project can be observed in real-world situations.

References

[1] Michael Biehl, Peter Riegler, and Christian Wöhler. "Transient dynamics of on-line learning in two-layered neural networks". In: *Journal of Physics A: Mathematical and General* 29.16 (1996), pp. 4769–4780. DOI: 10.1088/0305-4470/29/16/005. URL: https://doi.org/10.1088%2F0305-4470%2F29%2F16%2F005.

[2] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2011, pp. 315–323. URL: http://proceedings.mlr.press/v15/glorot11a.html.

[3] Michiel Straat and Michael Biehl. "On-line learning dynamics of ReLU neural networks using statistical physics techniques". In: *CoRR* abs/1903.07378 (2019). arXiv: 1903.07378. URL: http://arxiv.org/abs/1903.07378.

[4] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[5] Michiel Straat et al. "Statistical Mechanics of On-Line Learning Under Concept Drift". In: *Entropy* 20 (Oct. 2018), p. 775. DOI: 10.3390/e20100775.

[6] Michiel Straat et al. *Supervised Learning in the Presence of Concept Drift: A modelling framework*. 2020. arXiv: 2005.10531 [cs.LG].

[7] Michael Biehl and H Schwarze. "Learning by on-line gradient descent". In: *Journal of Physics A: Mathematical and General* 28 (Feb. 1995), pp. 643–656. DOI: 10.1088/0305-4470/28/3/018.

[8] Michiel Straat. "On-line learning in neural networks with ReLU activations". In: 2018.

[9] David Saad and Sara Solla. "On-line learning in soft committee machines". In: *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics* 52 (Nov. 1995), pp. 4225–4243. DOI: 10.1103/PhysRevE.52.4225.

[10] Gregory Ditzler et al. "Learning in Nonstationary Environments: A Survey". In: *Computational Intelligence Magazine, IEEE* 10 (Nov. 2015), pp. 12–25. DOI: 10.1109/MCI.2015.2471196.

[11] M Biehl and H Schwarze. "Learning drifting concepts with neural networks". In: *Journal of Physics A: Mathematical and General* 26.11 (1993), pp. 2651–2665. DOI: 10.1088/0305-4470/26/11/014. URL: https://doi.org/10.1088%2F0305-4470%2F26%2F11%2F014.

[12] M Biehl and H Schwarze. "On-Line Learning of a Time-Dependent Rule". In: *Europhysics Letters (EPL)* 20.8 (1992), pp. 733–738. DOI: 10.1209/0295-5075/20/8/012. URL: https://doi.org/10.1209%2F0295-5075%2F20%2F8%2F012.

[13] F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain". In: *Psychological Review* (1958), pp. 65–386.

[14] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539.