Bachelor's Thesis

Survey on Regularization Methods in Continual Learning

Department of Statistics
Ludwig-Maximilians-Universität München

Jörg Schantz

Munich, March 20th, 2025

Submitted in partial fulfillment of the requirements for the degree of B. Sc.
Supervised by Dr. Julian Rodemann

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. [4]

Contents

1 Introduction **1**

2 Neural Networks **1**

3 Framework **2**
3.1 Scenarios 2
3.2 Stability-Plasticity Trade-off 3

4 Metrics **4**

5 Regularization **5**
5.1 Regularization via Parameters 6
5.1.1 Quadratic Penalties 6
5.1.2 Sparse Penalties 10

6 Conclusion **10**

A Appendix **V**
A.1 Expansion of eq. 2 in [18] for T samples (20) V
A.2 Proof that CRR is biased V

B Electronic appendix **VII**

1 Introduction

Bli bla bulb

2 Neural Networks

Although continual learning is general modeling concept, applicable in statistical inference as well as pattern driven prediction algorithms, it is mostly used a in machine learning context. More specifically in artificial neural networks (ANN). They are algorithms based on the functionality of a human brain and often designed for scenarios where data is seen in real-time e.g. stock market predictions or power control systems.
The simplest form of an ANN is a single linear classifier, called one-neuron perceptron, that devides a vector x into two classes using a so-called activation function $h(\cdot)$ [8]. The neuron's input is given by

$$\sum_{i=1}^{n} w_i x_i + c = w^\top x + c \tag{1}$$

where n is the number of observations, w a weight vector assigned to x and c the decision threshold. The two class regions are separated by the hyperplane [8]

$$w^\top x + c = 0 \tag{2}$$

. Using multiple neurons with the same activation function creates a one-layer perceptron and enables classification for more than two classes with the input

$$\sum_{k=1}^{m} \sum_{i=1}^{n} w_{k,i} x_i + c = (w_1^\top x + c, ..., w_m^\top x + c)^\top = W^\top x + c \tag{3}$$

where W is the $n \times m$ weight matrix and m the number of classes. Given h the logistic function a one-layer perceptron is equal to a multinomial logit model [12]. Composing l layers of neurons, Feed Forward NN (FFNN), allows for a more and more abstract representation of the data and finer class boundaries. The unknown weight matrices $W_1, ..., W_l$ and the decision threshold c are the solution to the minimization problem

$$\hat{\theta} = \arg\min \sum_{i=1}^{n} L(f(x_i, \theta), y_i) \tag{4}$$

where θ are the unknown parameters, and $L(\cdot)$ a loss function which measures the difference between the predicted values $f(x_i, \theta)$ and true values y_i.

3 Framework

Throughout literature continual learning in a statistical sense means modeling a joint probability distribution $\mathbb{P}^{(T)}$, which is allowed to expand indefinitely [32]. T samples $D_t, t \in 1, ..., T$ from different distributions $\mathbb{P}_t$ are processed sequentially. A single sample has the form $D_t = (X^{(t)}, y^{(t)}) \in (\mathbb{R}^{n_t \times d_t}, \mathbb{R}^{n_t})$ with $X^{(t)}$ being the covariate matrix and $y^{(t)}$ the dependent variable. The D_t are assumed to be conditionally independent but not necessarily identically distributed [32]. Each tuple $(D_t, \mathbb{P}_t)$ may correspond to a distinct regression or classification task that is to be learned. The goal is to train a single model which is able to perform well on all tasks, although it is trained sequentially and cannot necessarily revisit prior tasks.

3.1 Scenarios

In regards to the distribution $\mathbb{P}$ of $Y^{(t)} = \{y^{(1)}, ..., y^{(t)}\}$ over which the model is evaluated after seeing the t-th samples, [6] and [32] differentiate between eight CL scenarios:
Task-incremental learning (TIL), *Class-incremental learning* (CIL), *Task-Free continual learning* (TFCL) and *Online contiunal learning* (OCL) algorithms all aim to learn a distinct set of tasks, while providing a task identity, if not stated otherwise [6, 32].

$$\emptyset = y^{(i)} \cap y^{(i+1)} \Rightarrow \mathbb{P}(Y^{(t+1)}) = \prod_{i=1}^{t+1} \mathbb{P}(y^{(i)}) \tag{5}$$

TIL allows task individual output layers or the training of separate models for each task. The challenge then is less about forgetting (subsection 3.2) but finding a healthy balance between predicting accuracy and model complexity [31].
CIL restricts this approach by only training one model, which is introduced stepwise to different classification tasks. CIL only provides task identity during training [31]. For example with samples t an agent learns to classify hats or gloves and with sample $t + 1$ shirts or pants. When testing, it is then also required to classify hats or shirts.
TFCL does not provide any task identity to the model and only focuses on labels [2].
OCL limits its sample sizes to one and focuses on real-time training [6, 32].
Domain-incremental learning (DIL) algorithms seek to learn multiple tasks that share the same label space [6]. For example first learning to drive during sunny weather and later

on while it is rainy.

$$y^{(i)} = y^{(i+1)} \nRightarrow \mathbb{P}(y^{(i)}) = \mathbb{P}(y^{(i+1)}) \tag{6}$$

One could view this as a version of task-incremental learning, where task identity is secondary as all tasks have the same data labels. Thus design based strategies to inhibit catastrophic forgetting are not possible [31].
Instance-incremental learning (IIL) algorithms learn one common task for all training samples [6, 32].

$$y^{(i)} = y^{(i+1)}, \mathbb{P}(y^{(i+1)}) = \mathbb{P}(y^{(i+1)}) \Rightarrow \mathbb{P}(Y^{(t)}) = \mathbb{P}(y^{(i)}) \tag{7}$$

This is a special case of DIL where a model learns the distribution of one "domain" while only ever accessing snippets of the total available data. For example each sample contains new real-world photographs of cats to classify. Assuming OCL only learns one task, OCL is a special case of IIL where every data point is seen in sequence.
Blurred Boundary continual learning (BBCL), in contrast to all others so far, allows partially overlapping label spaces [6, 32].
Continual Pre-training (CPT) aims to improve knowledge transfer with sequentially arriving pre-training data [6, 32].

3.2 Stability-Plasticity Trade-off

The challenge of continual learning is to strike a balance between stability and plasticity. Models should retain knowledge of past tasks, stability, while being flexible enough to incorporate information from new data, plasticity. The sequential training nature of CL changes the weights acquired form learning task A to accommodate for a new task B. This abrupt loss of information is called catastrophic forgetting [14, 24, 25, 27]. A naive approach to solving this dilemma would be storing and replaying data to the network with each training step. This is impractical because the amount of data needed to be stored is proportional to the number of tasks learned.
Evron et al. define forgetting as

$$F(t) = 1/t \sum_{i=1}^{t} \|X^{(i)} w^{(t)} - y^{(i)}\|^2 \tag{8}$$

. They have analyzed catastrophic forgetting in linear regression under the assumptions that values of X are bounded by 1, tasks are jointly realizable with a bounded (by 1) norm and there are more parameters than observations in each sample. Realizability assumes

the existence of true model weights s.t. $y = Xw$ [29]. This enables them to focus only on minimizing the distance between new and old model weights. In their work they find an upper bound for forgetting

$$\sup F(t) = \sup 1/t \sum_{i=1}^{t} \|(I - Q_i)Q_t...Q_1\|^2 \tag{9}$$

where Q_i are the projections onto the solution spaces of $w^{(i)}$ i.e. $Q_i := I - X^{(i)\top}(X^{(i)}X^{(i)\top})^{-1}X^{(i)}$. So far many methods of minimizing catastrophic forgetting have been developed. Their core ideas can be summarized to *Replay* methods [3, 7, 28], *Optimization* methods [16, 22, 26], *Architecual* methods [10, 13, 23] and *Regularization* methods, which will be discussed in section 5.

4 Metrics

Intro.
In the following each sample $D_t = (X^{(t)}, y^{(t)})$ is divided into a training split $D_t^{(train)} = (X_{(train)}^{(t)}, y_{(train)}^{(t)})$ and a testing split $D_t^{(test)} = (X_{(test)}^{(t)}, y_{(test)}^{(t)})$. The chosen splitting method is arbitrary. The training process for each sample will be conducted with $D_t^{(train)}$ and evaluation with $D_t^{(test)}$.
[32] mention different measures for model performance, stability and plasticity. I will focus on the dynamic forms given by [9], because they are adapted for in training use i.e. they represent a model's current state after the t-th training step.
Accuracy **A** represents a models performance i.e. how well the predictions $\hat{y}_{(test)}^{(t)}$ align with the true values of $y_{(test)}^{(t)}$ for a metric μ . When $A_{i,k}$ is the accuracy measured on the k-th test split after the i-th training step, then

$$\mathbf{A} = \frac{2}{t(t-1)} \sum_{i \geq k}^{t} A_{i,k} \tag{10}$$

is the average accuracy after the t-th training step over all test splits $D_k^{(test)}, k <= t$ [9].
Backward Transfer **BWT** evaluates a models stability [32]. The metric quantifies the influence of learning sample $D_{t+1}^{(train)}$ has on the performance over test sample $D_t^{(test)}$ [22]. Given, the above mentioned, individual *Accuracy* scores $A_{i,k}$

$$\mathbf{BWT} = \frac{2}{t(t-1)} \sum_{i=2}^{t} \sum_{k=1}^{i-1} (A_{i,k} - A_{k,k}) \tag{11}$$

is the average backward transfer after the t-th training step [9]. Note that **BWT** can be negative. This property captures (catastrophic) forgetting [32].
Forward Transfer **FWT** is a metric for model plasticity [32]. Complementary to BWT, *Forward Transfer* measures how previous training steps influence the current one. Again the individual *Accuracy* scores are the basis for this evaluation metric. The average influence of old training steps on the model performance after the t-th step:

$$\mathbf{FWT} = \frac{2}{t(t-1)} \sum_{i<k}^{t} A_{i,k} \tag{12}$$

[9].
Another metric that directly measures the relationship between stability and plasticity is presented in [26]. The authors use the maximum eigenvalue of the loss' Hessian λ^{max} to describe the width of their approximation of the loss' minimum. They hypothesize that the *wideness* of this minimum correlates with the forgetting rate of the respective model. Given $W^{(t)*}$ and $W^{(t+1)*}$ the optimal parameters after learning the t-th and $t+1$-th task and $L_t(\cdot)$ and $L_{t+1}(\cdot)$ the corresponding loss functions. Mirzadeh et al. formulate the upper bound

$$F_t = L_t(W^{(t+1)*}) - L_t(W^{(t)*}) \approx \frac{1}{2}\Delta W^\top \nabla^2 L_t(W^{(t)*})\Delta W \leq \frac{1}{2}\lambda_t^{max}\|\Delta W\|^2 \tag{13}$$

for the forgetting F_t of the t-th task. They approximate $L_t(W^{(t+1)*})$ around $W^{(t)*}$ with a second order Taylor approximation, where ∇^2 is the Hessian for L_t and ΔW the difference between $W^{(t+1)*}$ and $W^{(t)*}$. They argue that the loss can be approximated this way, because of its almost convex path around the minimum, for models that have more observations per sample than parameters.
Further, ΔW is dependent on the training process of the $t+1$-th task, which depends on the random sample it is trained on, so one can view the differences in parameters as a random vector, that follows some distribution parameterized by the eigenvalues of $\nabla^2 L_t(W^{(t)*})$ [26].

Controlling the distance of the weights seems to be the key to mitigating forgetting...

5 Regularization

As mentioned in subsection 3.2, one way to address the stability-plasticity problem is the use of regularization. This approach adds a penalty term to the loss function of a

model. Usually this penalty term depends on the model parameters. Later we will also see some methods that directly penalize the output of a model. I will begin by categorizing the regularization methods that I have found through out my research and present some selected examples. After this overview of current possibilities in regularization techniques and present attempts at unifying and generalizing them.
In [11, 19] the authors introduce a rudimentary approach to regularization in CL, *ordinary conitunal learning*. It is used as a worst case comparison for their contribution to this field and is the basis for the upper bound on forgetting in subsection 3.2.
Despite its limitations, I believe that it is a beneficial to start this section with ordinary CL. It reduces the complexity of neural networks to two linear regression problems, which makes for a softer entry into the field.
Assuming a CL problem with $T = 2$ linear regression models $y^{(t)} = X^{(t)}w^* + \epsilon_t, \epsilon_t \sim N(0, \sigma^2), t \in \{1, 2\}$, the task corresponding samples $D_t = (X^{(t)}, y^{(t)}) \in (\mathbb{R}^{n_t \times d}, \mathbb{R}^{n_t})$ and the commutable covariance matrices Σ_t. Then ordinary continual learning algorithm performs an ordinary least square (OLS) minimization over the first sample set D_1 to estimate the parameters $\hat{w}^{(1)} = (X^{(1)\top}X^{(1)})^{-1}X^{(1)\top}y^{(1)}$. In the second training sequence, ordinary CL fits $w^{(2)}$ to the residuals of task one with respect to $X^{(2)}$. The new parameters $\hat{w}^{(2)}$ are then:

$$\hat{w}^{(2)} = \hat{w}^{(1)} + (X^{(2)\top}X^{(2)})^{-1}X^{(2)\top}(y^{(2)} - X^{(2)}\hat{w}^{(1)}) \tag{14}$$

. In their analysis of ordinary continual learning, [19] show that it suffers from catastrophic forgetting when dealing with "dissimilar" tasks i.e. similarity is measured via the following bound:

$$d_F \leq tr(\Sigma_1\Sigma_2^{-1}) \tag{15}$$

where d_F is the expected forgetting rate between the two tasks.
Due to the heavy assumptions made, especially that both minimization problems have the common solution w^*, ordinary CL is only applicable in DIL and IIL. This highlights the need for methodologies that control weight deviation when trying to combat forgetting in a less restrictive setting.

5.1 Regularization via Parameters

5.1.1 Quadratic Penalties

Expanding on the naive *ordinary continual learning* approach, Li et al. suggest an adaptation of the Ridge penalty [12] for continual learning, dubbed *continual ridge regression* (CRR) [19, 36]. Again the $\hat{w}^{(t)}$ follow a Gaussian distribution $N(w^*, \sigma^2(X^{(t)\top}X^{(t)})^{-1})$ and the estimation of $w^{(1)}$ stays the same as in ordinary CL, too. For estimating $w^{(2)}$, they

introduce now a ridge-like penalty term

$$\text{pen}(w) = \lambda \|w - \hat{w}^{(1)}\|_2^2 \tag{16}$$

which centers the new weights around the previously estimated $\hat{w}^{(1)}$ instead of 0. Instead of using penalized least squares, the authors decide to perform a penalized mean squared error regression.

$$\begin{aligned} \hat{w}^{(2)} &= \arg\min_w \frac{1}{n} \|y^{(2)} - X^{(2)} w\|_2^2 + \text{pen}(w) \\ &= (X^{(2)\top} X^{(2)} + \lambda n I)^{-1} (X^{(2)\top} y^{(2)} + \lambda n \hat{w}^{(1)}) \end{aligned} \tag{17}$$

Like regular ridge regression, CRR is also biased for $\lambda \neq 0$. The distribution of $\hat{w}^{(2)}$ is also Gaussian with $\mathbb{E}(\hat{w}^{(2)}) = A(\lambda n \hat{w}^{(1)} + X^{(2)\top} X^{(2)} \mathbb{E}(w^{(2)}))$ and $\mathbb{V}(\hat{w}^{(2)}) = \sigma_2^2 A^\top X^{(2)\top} X^{(2)} A$, where $A = (X^{(2)\top} X^{(2)} + \lambda n I)^{-1}$. The proofs for a biased CRR and its distributional properties can be found in subsection A.2. The authors of [19] acknowledge that centering around $\hat{w}^{(1)}$ enables a more stable learning environment, compared to ordinary CL, but still struggles when tasks are too dissimilar. Another reason is that all dimensions of w are penalized equally throughout all tasks, i.e. $\text{pen}(w) = (w - \hat{w}^{(1)})^\top \lambda I (w - \hat{w}^{(1)})$. When learning a joint probability distribution, as in DIL, the information contained in D_t about w can vary across coordinates. As a solution to this, [36] propose a generalized quadratic penalty for linear regression tasks, which allows individual regularization strengths in all directions of w.

Generalized l2-regression (GR) [36] extends CRR and ordinary CL to more than $T > 2$ tasks and is asymptotically equivalent to an unrestricted model, i.e. a model that can access all data samples at the same time. They state that the unrestricted estimation error $\mathcal{L}(\cdot)$ over all tasks is

$$\mathcal{L}(\hat{w}^{(T)}) = \sum_i^d \frac{\sigma^2}{\alpha_i^{(1)} n^{()} + ... + \alpha_i^{(T)} n^{(T)}} \tag{18}$$

with $\alpha_i^{(t)}$ being the i-th eigenvector of Σ_t. Note that this error is monotonously decreasing as T gets bigger thus no forgetting [36]. GRs goal now, is to find a matrix $\Lambda^{(t)}$ which properly accommodates a samples contribution to each $\hat{w}_i$ so that the combined estimation error of $L(\hat{w}^{(t)})$ converges to $\mathcal{L}(\hat{w}^{(T)})$. The authors are able to prove that for $\Lambda^{(t)}$ a diagonizable matrix with $\Lambda^{(t)} = U \Delta U^\top, \Delta = \text{diag}(\delta_1, ...\delta_d)$ then $L(\hat{w}^{(t)})$ is bounded by

$\mathcal{L}(\hat{w}^{(T)})$ if the diagonal values of Δ are

$$\delta_i = \frac{\sigma^2/(U_i^\top w^*)^2 + \alpha_i^{(1)} n^{(1)} + ... + \alpha_i^{(t-1)} n^{(t-1)}}{n^{(t)}} \tag{19}$$

. For large $n^{(t)}$, $\frac{\sigma^2/(U_i^\top w^*)^2}{n^{(t)}}$ becomes small enough to be dropped and [36] approximate $\tilde{\Lambda}^{(t)} = \frac{1}{n^{(t)}} \sum_{i=1}^{t-1} n^{(i)} \Sigma_i$.
In this linear setting, the Σ_i are equivalent to the Hessian and Fisher information matrix (FIM) of the loss function. This means that every w_i penalized proportionally to the information sample D_t provides contains about it.
[36] demonstrate how powerful regularization can be. Though, a shared label space for y is not always realistic, see CIL. [18] tackle this problem by taking a Bayesian look at the joint distribution over all tasks.
One of the most influential regularization approaches for CL is the *elastic weight consolidation* penalty (EWC) by Kirkpatrick et al. [5, 15, 19, 21, 30, 33, 35, 36]. They suggest measuring weight importance via the Fisher information matrix (FIM) [18]. Kirkpatrick et al. justify this approach through a probabilistic view of neural networks. They no longer want to find the parameters that best fit the data pattern but find the most probable model weights, depending on a given data sample. Using Bayes' Rule and the assumption of independent samples (e.g. CIL), they express the conditional probability $\mathbb{P}(w|\mathcal{D}^{(t)}), \mathcal{D}^{(t)} = \{D_1, ..., D_t\}$ of the weights as

$$\log(\mathbb{P}(w|\mathcal{D}^{(t)})) = \log(\mathbb{P}(D_t|w)) + \log(\mathbb{P}(w|\mathcal{D}^{(t-1)})) - \log(\mathbb{P}(D_t)) \tag{20}$$

and point out that all of the information about all prior tasks is in $\mathbb{P}(w|\mathcal{D}^{(t-1)})$, which is unavailable due to the sequential training constraint. To overcome this problem, the authors approximate the missing posterior as a Gaussian with expected value $\hat{w}^{(t-1)}$ and precision matrix $F = diag(\sum_{i<t} \mathcal{I}_i(w_1), ..., \sum_{i<t} \mathcal{I}_i(w_d))$ where $\mathcal{I}_i(w_j), j \in \{1, ..., d\}$ are the Fisher information of w_i from the i-th training step, thus $\mathbb{P}(w|\mathcal{D}^{(t-1)}) \dot{\sim} N(\hat{w}^{(t-1)}, F^{-1})$.
The resulting penalty function is a weighted Ridge penalty, where the squared deviation from the previous parameters is weighed against its Fisher information:

$$\text{pen}_t(w) = \frac{\lambda}{2}(w - \hat{w}^{(t-1)})^\top F (w - \hat{w}^{(t-1)}) \tag{21}$$

Note that if the loss is chosen as the negative log-likelihood, EWC is equal to the 2nd Taylor approximation of a generalized forgetting rate in (13). In this case the FIM is equivalent to the Hessian of the loss. In general the EWC penalty encourages gradient decent to follow along trajectories of w_i with low FI which are thus less prone to forget-

ting.
Neural networks often rely on numerical gradient decent solutions for parameter estimation [12], Zenke et al. argue in [35] that a static estimate of parameter importance between training steps is not enough and suggest a dynamic solution, *synaptic intelligence* (SI), along the loss' gradient. Similar to EWC and CRR they impose a quadratic penalty on the loss:

$$\text{pen}(w) = \frac{\lambda}{2}(w - \hat{w}^{(1)})^\top H(w - \hat{w}^{(1)}) \tag{22}$$

where H is the diagonal of the Hessian of the current loss $L(X^{(2)}, y^{(2)}, w)$. Approximating the true Hessian with its diagonal entries, again implies independent covariates like in EWC and CRR.
The next example is the *memory aware synapses* (MAS) penalty [1]. Similar to EWC and SI it focuses on task disjoint CL. To further improve the idea of gradient based importance, the authors consider the model output $h(\cdot)$ and measure its sensitivity to changes in the model parameters. The importance matrix Ω holds the mean gradients over all samples of the squared L2-normed outputs i.e. $\Omega^{(t)} = 1/n \sum_{i \leq n} \nabla \|h(x_i^{(t)}, \hat{w}^{(t)})\|_2^2$. With the resulting penalty function:

$$\text{pen}_t(w) = \lambda(w - \hat{w}^{(t-1)})^\top \Omega^{(t-1)}(w - \hat{w}^{(t-1)}) \tag{23}$$

. [1] aim to provide flexible importance measure, which can be calculated on any representative data set, since it does not depend on the model loss.

Generally regularization in CL through a squared penalty restricts large deviations from the a fore estimated parameters if these $w_i^{(t-1)}$ were important to prior learnings. Given some importance matrix A a general l2-penalty for the next training step is:

$$pen_{t+1}(w) = \lambda(w - \hat{w}^{(t)})^\top A(w - \hat{w}^{(t)}) \tag{24}$$

. Although FIM and Hessian at large are not identical, [5] demonstrate how SI and MAS are still linked to FIM. [33] provide a unifying analysis of squared penalties in CL. They conclude that the difference between the true loss over all tasks and its approximation depends on two factors. First a sample effect, which is negligible for increasing $n^{(t)}$, and second the technical of the approximation, thus call for more accurate approximations. Furthermore [20] point out that the diagonal approximation of the FIM has potential to lead gradient decent "off-path" and use rotations of the parameter space to adjust.

5.1.2 Sparse Penalties

The ridge like penalties provide control over a model's stability. Because their approximations become increasingly inaccurate with each new training step, they encourage smaller steps away from the current state of the model as training continues [33]. All of the algorithms presented so far, imply that every parameters are, to some degree, useful across all tasks. [17, 34] question this and suggest a Grouped-LASSO [12] penalty. The parameter groups are determined by the neurons they connect to or come from. This way the model can benefit from already established weights and simultaneously use free neurons to fit task specific parameters. In order to protect task individual neurons, [34] rely on multiple retraining of the same model to identify "empty" neurons.

6 Conclusion

Blub bla bli

A Appendix

A.1 Expansion of eq. 2 in [18] for T samples (20)

Let $D_i, i \in \{1, ..., t\}$ be t independent samples, as described in section 3, $\mathcal{D}^{(t)} = \{D_1, ..., D_t\}$ the joint samples and $w \in \mathbb{R}^d$ a weight vector. Then the conditional probability

$$\begin{aligned} \mathbb{P}(\mathcal{D}^{(t)}|w) &= \frac{\mathbb{P}(D_1, ..., D_t, w)}{\mathbb{P}(w)} \\ &= \frac{\mathbb{P}(D_1, ..., D_{t-1}|D_t, w)\mathbb{P}(D_t, w)}{\mathbb{P}(w)} \\ &= \mathbb{P}(D_1, ..., D_{t-1}|w)\mathbb{P}(D_t|w) \end{aligned} \tag{25}$$

This we plug into the Bayes' Rule for the posterior $\mathbb{P}(w|\mathcal{D}^{(t)})$ and get

$$\begin{aligned} \mathbb{P}(w|\mathcal{D}^{(t)}) &= \frac{\mathbb{P}(\mathcal{D}^{(t)}|w)\mathbb{P}(w)}{\mathbb{P}(\mathcal{D}^{(t)})} \\ &= \frac{\mathbb{P}(D_1, ..., D_{t-1}|w)\mathbb{P}(D_t|w)\mathbb{P}(w)}{\mathbb{P}(\mathcal{D}^{(t)})} \\ &= \frac{\mathbb{P}(w|D_1, ..., D_{t-1})\mathbb{P}(D_t|w)}{\mathbb{P}(D_t)} \end{aligned} \tag{26}$$

The approximate Gaussian for the posterior $\mathbb{P}(w|D_1, ..., D_{t-1})$ of all prior tasks is then $N(w, (\sum_{i=1}^{t-1} \text{diag}(F_i))^{-1})$ using the chain rule for independent Fisher information $F_i = \mathcal{I}_{D_i}(w)$.

A.2 Proof that CRR is biased

Let $D_1 = (X_1, y_1)$ and $D_2 = (X_2, y_2)$ be two independent linear regression problems, with $y_i \sim N(X_i w_i, \sigma_i^2 I), i \in \{1, 2\}$. Begin with matrix form of $\hat{w_2}$:

$$\begin{aligned} \hat{w_2} &= \arg\min_{w_2} \frac{1}{n}\|y_2 - X_2^\top w_2\|_2^2 + \text{pen}(w_2) \Leftrightarrow \\ 0 &= \nabla[\frac{1}{n}\|y_2 - X_2^\top w_2\|_2^2 + \text{pen}(w_2)] \\ &= \nabla[\frac{1}{n}(y_2 - X_2 w_2)^\top (y_2 - X_2 w_2) + \lambda(w_2 - \hat{w_1})^\top (w_2 - \hat{w_1})] \\ &= -\frac{2}{n}X_2^\top Y_2 + \frac{2}{n}w_2 X_2^\top X_2 + 2\lambda w_2 - 2\lambda\hat{w_1} \\ \hat{w_2} &= (X_2^\top X_2 + \lambda n I)^{-1}(X_2^\top y_2 + \lambda n \hat{w_1}) \end{aligned} \tag{27}$$

Continue with $\mathbb{E}[\hat{w_2}]$. Define $A := (X_2^\top X_2 + \lambda n I)^{-1}$.

$$\begin{aligned}\mathbb{E}[\hat{w_2}] &= \mathbb{E}[(X_„^\top X_2 + \lambda n I)^{-1}(X_2^\top y_2 + \lambda n \hat{w_1})] \\ &= A(\lambda n \hat{w_1} + \mathbb{E}[X_2^\top (X_2 w_2 + e_2)]) \\ &= A\lambda n \hat{w_1} + A X_2^\top X_2 \mathbb{E}[w_2] \\ &\neq \mathbb{E}[w_2]\end{aligned} \tag{28}$$

For $\lambda = 0 \Rightarrow A = (X_2^\top X_2)^{-1} \Rightarrow \mathbb{E}(\hat{w_2}) = \mathbb{E}(w_2)$.

End with $\mathbb{V}(\hat{w_2})$

$$\begin{aligned}\mathbb{V}(\hat{w_2}) &= \mathbb{V}(A(X_2^\top y_2 + \lambda n \hat{w_1})) \\ &= \mathbb{V}(y_2) A^\top X_2^\top X_2 A \\ &= \sigma_2^2 I A^\top X_2^\top X_2 A\end{aligned} \tag{29}$$

.

B Electronic appendix

Data, code and figures are provided in electronic form.

References

[1] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget, 2018. URL https://arxiv.org/abs/1711.09601.

[2] R. Aljundi, K. Kelchtermans, and T. Tuytelaars. Task-free continual learning, 2019. URL https://arxiv.org/abs/1812.03596.

[3] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning, 2019. URL https://arxiv.org/abs/1903.08671.

[4] S. H. Bach and M. A. Maloof. *A Bayesian Approach to Concept Drift*, pages 127–135. 2010.

[5] F. Benzing. Unifying regularisation methods for continual learning, 2021. URL https://arxiv.org/abs/2006.06357.

[6] S. A. Bidaki, A. Mohammadkhah, K. Rezaee, F. Hassani, S. Eskandari, M. Salahi, and M. M. Ghassemi. Online continual learning: A systematic literature review of approaches, challenges, and benchmarks, 2025. URL https://arxiv.org/abs/2501.04897.

[7] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. S. Torr, and M. Ranzato. On tiny episodic memories in continual learning, 2019. URL https://arxiv.org/abs/1902.10486.

[8] K.-L. Du and M. N. S. Swamy. *Neural Networks and Statistical Learning*. Springer London, 2 edition, 2019.

[9] N. Díaz-Rodríguez, V. Lomonaco, D. Filliat, and D. Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning, 2018. URL https://arxiv.org/abs/1810.13166.

[10] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach. Adversarial continual learning, 2020. URL https://arxiv.org/abs/2003.09553.

[11] I. Evron, E. Moroshko, R. Ward, N. Srebro, and D. Soudry. How catastrophic can catastrophic forgetting be in linear regression?, 2022. URL https://arxiv.org/abs/2205.09588.

[12] L. Fahrmeir, T. Kneib, S. Lang, and B. D. Marx. *Regression - Models, Methods and Applications*. Springer Berlin, 2 edition, 2022.

[13] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra. Pathnet: Evolution channels gradient descent in super neural networks, 2017. URL https://arxiv.org/abs/1701.08734.

[14] R. M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. ISSN 1364-6613. doi: https://doi.org/10.1016/S1364-6613(99)01294-2. URL https://www.sciencedirect.com/science/article/pii/S1364661399012942.

[15] F. Huszár. Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences*, 115(11), Feb. 2018. ISSN 1091-6490. doi: 10.1073/pnas.1717042115. URL http://dx.doi.org/10.1073/pnas.1717042115.

[16] K. Javed and M. White. Meta-learning representations for continual learning, 2019. URL https://arxiv.org/abs/1905.12588.

[17] S. Jung, H. Ahn, S. Cha, and T. Moon. Continual learning with node-importance based adaptive group sparse regularization, 2021. URL https://arxiv.org/abs/2003.13726.

[18] J. Kirkpatrick, R. Pascanu, N. Rabinowitza, J. Veness, A. A. R. Guillaume Desjardins, K. Milan, J. Quan, T. Ramalho, D. H. Agnieszk Grabska-Barwinska, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *arXiv:1612.00796v2*, 2017.

[19] H. Li, J. Wu, and V. Braverman. Fixed design analysis of regularization-based continual learning, 2024. URL https://arxiv.org/abs/2303.10263.

[20] X. Liu, M. Masana, L. Herranz, J. V. de Weijer, A. M. Lopez, and A. D. Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting, 2018. URL https://arxiv.org/abs/1802.02950.

[21] N. Loo, S. Swaroop, and R. E. Turner. Generalized variational continual learning, 2020. URL https://arxiv.org/abs/2011.12328.

[22] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning, 2022. URL https://arxiv.org/abs/1706.08840.

[23] A. Mallya, D. Davis, and S. Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights, 2018. URL https://arxiv.org/abs/1801.06519.

[24] J. Mcclelland, B. Mcnaughton, and R. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102 3:419–457, 1995. doi: https://doi.org/10.1037/0033-295X.102.3.419.

[25] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

[26] S. I. Mirzadeh, M. Farajtabar, R. Pascanu, and H. Ghasemzadeh. Understanding the role of training regimes in continual learning, 2020. URL https://arxiv.org/abs/2006.06958.

[27] R. Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological RReview*, 97 2:285–308, 1990. URL https://api.semanticscholar.org/CorpusID:18556305.

[28] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning, 2017. URL https://arxiv.org/abs/1611.07725.

[29] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learing: From Theory to Algorithms*. Cambridge University Press, 2014.

[30] M. K. Titsias, J. Schwarz, A. G. de G. Matthews, R. Pascanu, and Y. W. Teh. Functional regularisation for continual learning with gaussian processes, 2020. URL https://arxiv.org/abs/1901.11356.

[31] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, Dec 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00568-3. URL https://doi.org/10.1038/s42256-022-00568-3.

[32] L. Wang, X. Zhang, H. Su, J. Zhu, Fellow, and IEEE. A comprehensive survey of continual learning: Theory and method and application, 2024. URL https://arxiv.org/abs/2302.00487.

[33] D. Yin, M. Farajtabar, A. Li, N. Levine, and A. Mott. Optimization and generalization of regularization-based continual learning: a loss approximation viewpoint, 2021. URL https://arxiv.org/abs/2006.10974.

[34] J. Yoon, E. Yang, J. Lee, and S. J. Hwang. Lifelong learning with dynamically expandable networks, 2018. URL https://arxiv.org/abs/1708.01547.

[35] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence, 2017. URL https://arxiv.org/abs/1703.04200.

[36] X. Zhao, H. Wang, W. Huang, and W. Lin. A statistical theory of regularization-based continual learning, 2024. URL https://arxiv.org/abs/2406.06213.

Declaration of authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, March 20th, 2025

Name