Project Machine Learning
— Milestone 1 —

Cederic Aßmann, Friedrich Hagedorn, Jakub Sliwa

November 20, 2024

1 Introduction

The main objective of our participation in the *Machine Learning Project* during the Winter Semester 2024/25 is to reconstruct the results achieved by Ilse et al. (2018) and published in the paper titled *Attention-based Deep Multiple Instance Learning*. The replication process has been divided into three milestones. Hence, this report aims to present our progress in achieving the final goal within the confines of the first milestone. The tasks within its scope include dataset curation, data visualization, model prototyping, and the evaluation of the baseline method[1].

The structure of this report is organised according to the following outline: Section 1 contains a description of the report's purpose and structure, as well as a statement of the Multiple Instance Learning (MIL) problem. In section 2, the dataset curation process is presented, including the characterization of the MNIST-Bags dataset with examples, data transformation, feature extraction method, and a discussion of what may be learned from the data. Section 3 introduces MIL pooling operators and various approaches to solving the MIL problem, including the attention-based embedding-level method, which is the main contribution of Ilse et al. (2018). Moreover, it features the optimization objective and the chosen validation metrics used to evaluate the baseline method's performance on the MNIST-Bags dataset. Lastly, it contains a detailed description of the conducted experiments and achieved results. Section 4 discusses challenges and possibilities offered by the MNIST-Bags dataset, as well as possible business applications. Additionally, we assess the progress made and outline future milestones. Lastly, the exact quantitative results of the experiments carried out can be found in the corresponding tables in Appendix A.

1.1 MIL problem statement

In a traditional supervised binary classification problem, each instance is assigned a corresponding class label that a machine learning model learns to predict for unseen data points. However, in a multiple instance learning (MIL) problem, individual instances are grouped together to form bags of instances, whose sizes may vary. Each bag's binary label is determined by the labels of the instances it contains, even though these individual instance labels are not accessible during model training. Consequently, a MIL model attempts to learn to predict bag labels for bags not seen during training. Moreover, it must be permutation-invariant, i.e. it must disregard the ordering of individual instances within a bag. MIL problems often arise in complex machine learning applications, such as medical imaging, where instance-level labels are unavailable or prohibitively expensive to obtain.

A suitable example of a MIL problem is provided by Dietterich et al. (1997). Individual instances are keys that either can or cannot open the supply room door. Bags are represented by key chains, consisting of a varying number of individual keys. A bag's label indicates whether any of the keys it contains can open the supply room door. In such a case, a MIL model would learn to predict whether an unseen key chain is able to unlock the door or not.

[1] The full implementation is available at `https://git.tu-berlin.de/cederic/attdmil`.

2 Dataset overview

2.1 MNIST-Bags

The well-known MNIST dataset is a collection of images representing hand-written digits from 0 to 9. It consists of 60.000 training and 10.000 test samples, each with a size of 28x28 pixels, where each cell holds a floating point value between 0.0 and 1.0, corresponding to its color on the greyscale[2]. No image contains any missing values.
MNIST-Bags dataset is therefore a collection of bags, each of which consists of a random number of instances from the MNIST dataset. The size of a bag is drawn from the normal distribution with given parameters (e.g. $\mathcal{N}(10, 2)$) and rounded to the nearest integer. Naturally, the minimal bag size is 1.
A label of a bag is determined by a presence of at least one instance representing a chosen digit. In compliance with the paper Ilse et al. (2018), a bag is assigned a positive label if it contains one or more instances depicting '9', and a negative one if it does not. Fig. 1 represents an example of a negative bag. In Fig. 2, an example of a positive bag with a single '9' is given. Fig. 3 shows a positive bag with multiple instances of '9'. Note that each bag has a different size.

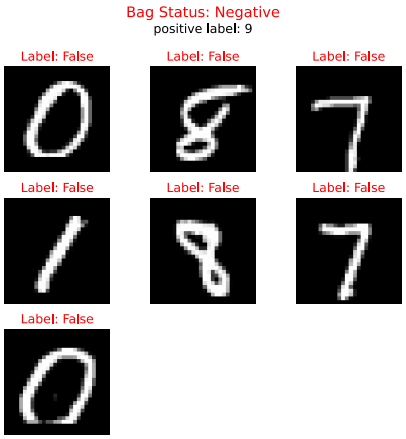

Figure 1: Example of a negative bag.

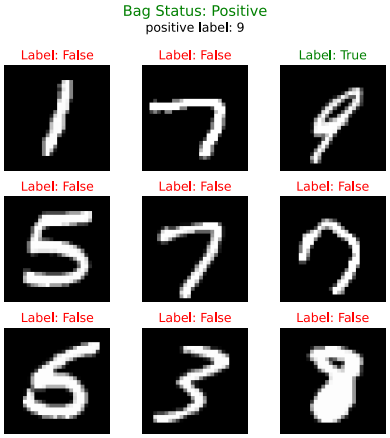

Figure 2: Example of a positive bag with a single instance of '9'.

2.2 Data preprocessing

Normalization In order to accelerate the training process and improve model's convergence, the input features were normalized based on the MNIST dataset's distribution. As a result, the input features are centered around 0 with a standard deviation of 1.

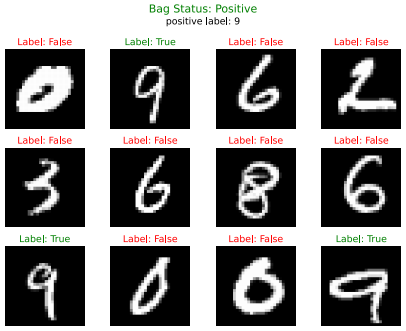

Figure 3: Example of a positive bag with multiple instances of '9'.

Feature extraction In compliance with Ilse et al. (2018), a variation of LeNet-5 model is used as a feature extraction method in all approaches. The convolutional layers are responsible for capturing features, starting from simple patterns to more complex, hierarchical ones as the network goes deeper. Subsampling layers, such as pooling layers, reduce the spatial dimensionality of feature maps, making the network more efficient and robust while retaining essential information. In comparison to the original LeNet-5 architecture, average-pooling was replaced with max-pooling, sigmoidal activation was substituted with ReLU non-linearity and only one fully-connected layer was used Lecun et al. (1998). Tab. 1 presents the architecture of the obtained feature extractor.

[2]Originally, the MNIST dataset contains integer values from 0 to 255, but they were scaled to fit into the [0, 1] range.

Layer	Type
1	conv(5,1,0)-20 + ReLU
2	maxpool(2,2)
3	conv(5,1,0)-50 + ReLU
4	maxpool(2,2)
5	fc-500 + ReLU

Table 1: Architecture of the adjusted LeNet-5 feature extractor.

Dataset balancing Our implementation of the MNIST-Bags ensures that the obtained dataset is perfectly balanced, i.e. exactly half of the bags are given a positive label. This applies to both training and testing dataset. The rationale behind it is an attempt to prevent the model from underperforming on negative bags for larger bag sizes (e.g. 100), when the likelihood of creating a bag excluding '9' is extremely small[3].

Duplicates Even though the MNIST dataset does not contain any duplicates, it is still theoretically possible for two bags to consist of the same instances. This occurs because, during bag creation, instances are sampled with replacement, meaning each instance may be included in multiple different bags[4]. However, the likelihood of such an event occurring is marginal and can therefore be ignored.

Dataset storage Due to the diverse experimental settings involving varying mean bag sizes and numbers of training bags, we decided not to store each individual MNIST-Bags dataset on the cluster[5]. Instead, every time the training is initiated, the MNIST dataset is loaded from our storage path and the bag creation process is executed. To ensure replicability, i.e. producing the same MNIST-Bags dataset across runs, a seed for random number generation has been implemented.

Input shape The model's ability to handle inputs of varying shapes depends on what kind of input is considered. While the model can naturally process bags containing different numbers of instances, it cannot handle variations in the shapes of individual instances. In theory, the model can process two bags simultaneously; however, following the approach of Ilse et al. (2018), a batch size of 1 is used, meaning the model processes one bag at a time.

2.3 Learning about MNIST-Bags

Machine learning algorithms trained on the MNIST-Bags dataset can be used to solve the MIL problem, i.e. learn the Bernoulli distribution of the bag label. Then, these algorithms are able to accurately predict the labels of unseen bag instances.
Moreover, some approaches may also enable determining each image's contribution to the bag label. In such cases, it is possible to identify key instances in the bag and infer, whether individual instance represents '9' or not.
Lastly, unsupervised methods may be used to detect similarities between different bags. This would enable grouping of the bags, for example based on whether they contain instances representing the same subset of digits.

[3]The distribution of classes in the MNIST dataset is roughly uniform, meaning that each digit appears approximately the same number of times in both training and testing dataset. Therefore, the likelihood of creating a negative bag, with a mean bag size set to 100, amounts to $0.9^{100} \approx 0.0000027$.

[4]If sampling without replacement was used instead, some experiments with larger mean bag sizes and greater numbers of training bags could not be conducted under the current strategy of yielding a perfectly balanced dataset.

[5]There are 21 different MNIST-Bags datasets for all pairs of mean bag size and number of training bags.

3 Baseline method and evaluation

To begin with, techniques to solve the MIL problem can be separated into two main approaches which fulfill Theorems 1 and 2 from Ilse et al. (2018) - the instance-level approach and the embedding-level approach, which will be further explained in Sec. 3.2. Depending on the chosen approach the final layers of the neural network are connected differently. Regarding the possible MIL pooling operations, MIL pooling with the maximum operator and MIL pooling with the mean operator are the baseline operators used in previous works. The maximum operator is more widely used in research Feng and Zhou (2017); Pinheiro and Collobert (2015); Zhu et al. (2017) due to a better bag level classification performance Wang et al. (2016). Nevertheless, both pooling operations are non-trainable, which is why the authors Ilse et al. (2018) proposed the novel attention-based MIL pooling.

3.1 MIL pooling

MIL max pooling The MIL pooling operation needs to be permutation-invariant, as highlighted in the MIL problem statement 1.1. The maximum operator ensures that the score function $\hat{\theta}(X)$, more specifically the bag label prediction, is symmetric. This means that the order of elements in the tensor does not affect the outcome of this operation. The operation is defined as follows: $\forall d = 1, \ldots, d_{\text{feat}} : \; z_d = \max_{k=1,\ldots,K}\{\mathbf{h}_{kd}\}$ where $\mathbf{h} \in \mathbb{R}^{K \times d_{\text{feat}}}$ is a matrix and $\mathbf{z} \in \mathbb{R}^{d_{\text{feat}} \times 1}$ is a vector.

MIL mean pooling As stated above, the mean pooling operator fulfills the permutation-invariance criterion as well and is defined as follows: $\forall d = 1, \ldots, d_{\text{feat}} : \; z_d = \frac{1}{K} \sum_{k=1}^{K} \mathbf{h}_{kd}$ whereas $\mathbf{h} \in \mathbb{R}^{K \times d_{\text{feat}}}$ is a matrix and $\mathbf{z} \in \mathbb{R}^{d_{\text{feat}} \times 1}$ is a vector.

Attention-based MIL pooling The authors claim that the MIL max/mean pooling operators have a significant limitation: they are both predefined and non-trainable. As a result, they suggest that a flexible and adaptive MIL pooling method, one that can adjust to a specific task and data, could lead to better performance. Moreover, such an adaptive pooling method should ideally be interpretable, a quality that is currently lacking in both aforementioned operators. Hence, they propose the attention mechanism, previously used in Xu et al. (2016); Lin et al. (2017); Bahdanau et al. (2016); Raffel and Ellis (2016). It is suggested to use it for solving the MIL problem, in which instances are aggregated through a weighted average of their embeddings: $\mathbf{z} = \sum_{k=1}^{K} a_k \mathbf{h}_k$ where $\mathbf{h}_k$ denotes an embedding of one instance from a bag of instances. The authors propose to use a two-layered neural network, which calculates the attention weights a_k for K instances in one bag. Referring to Eq. 8 in Ilse et al. (2018), the a_k are calculated by a linear operation followed by an element-wise non-linear $\tanh(\cdot)$ and another linear operation. These linear operations include the usage of trainable/adjustable parameters. The output is then scaled by a softmax operation to the range $[0, 1]$. Invariance of a_k to the size of the bag is assured by: $\sum_{k=1}^{K} a_k = 1$.

Gated attention-based MIL pooling The attention mechanism can be extended by the gated-attention mechanism proposed originally by Dauphin et al. (2017). It differs only in the way, the attention weights a_k are calculated. This extension, referring to Eq. 9 in Ilse et al. (2018), includes the sigmoid non-linearity and another linear operation, so that both non-linear functions are aggregated. This should strengthen the learnable relations between instances in a bag, since the sigmoid function adds more non-linearity around the origin.

3.2 MIL approaches

In the following sections, the instance-level approach combined with the MIL mean/max operator is defined, as well as the embedding-level approach coupled with both the MIL mean/max operator and the proposed attention/gated-attention MIL operator is formulated. All approaches rely on the LeNet-5 feature extractor Lecun et al. (1998). Dimensions in the model overviews are specific to the MNIST-Bags dataset, but the core concepts of the architectures are applicable to any dataset.

The instance-level approach Given a bag of instances $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ the feature extractor calculates the corresponding bag of embeddings $H = \{\mathbf{h}_1, \ldots, \mathbf{h}_K\}$. Each embedding is the input to an instance-level classifier (one fully-connected layer), which reduces it to a single instance score. MIL mean/max pooling is further performed on the aggregated instance scores to obtain the final prediction $\hat{\theta}(X)$ of the single class label of the bag. The instance labels remain unknown, which Wang et al. (2016) proved to be the reason why this approach introduces some additional error to the single class label prediction. On the other hand, the instance-level approach provides interpretability, given the single instances scores. A detailed overview of this approach is provided in Fig. 4.

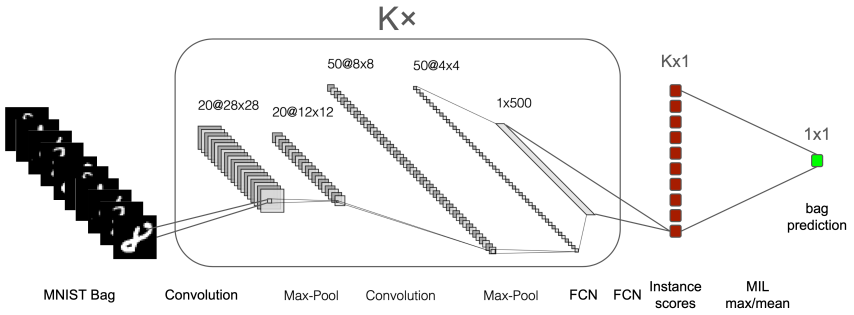

Figure 4: **Instance-level approach:** For each $\mathbf{x}_k$ in one MNIST-Bag the feature extractor creates an embedding $\mathbf{h}_k^{1\times d_{\text{feat}}}$. The instance scores received from the instance-level classifier are shown in red, whereas the final bag class prediction $\hat{\theta}(X)$ is highlighted in green.

The embedding-level approach Given the same architecture for feature extraction, the two approaches do not differ until the embeddings $H = \{\mathbf{h}_1, \ldots, \mathbf{h}_K\}$. The embedding approach, see Fig. 5, concatenates all $\mathbf{h}_k$ to a matrix $\mathbf{H}^{K\times d_{\text{feat}}}$. Using the previously stated mean/max pooling for matrices (Sec. 3.1), a bag representation $\mathbf{z} \in \mathbb{R}^{1\times d_{\text{feat}}}$ that is independent of the number of instances in the bag is calculated. Because of this operations, the embedding-level approach is not able to provide a single score for each instance, which results in a lack of interpretability for this approach. The bag's representation $\mathbf{z}$ is subsequently forwarded into a bag-level classifier (again one fully-connected layer) to provide the final prediction $\hat{\theta}(X)$ of this bag's class.

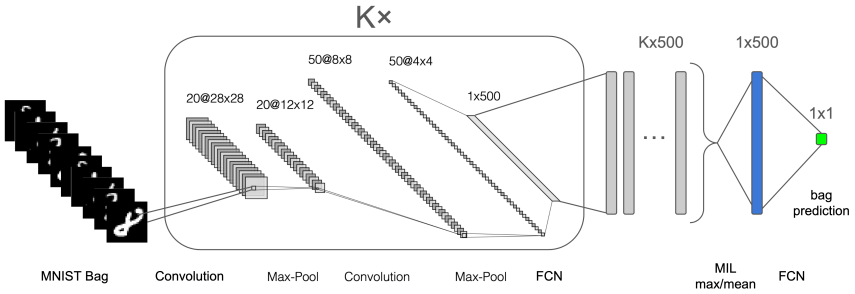

Figure 5: **Embedding-level approach:** For each $\mathbf{x}_k$ in one MNIST-Bag the feature extractor creates an embedding $\mathbf{h}_k^{1\times d_{\text{feat}}}$. The concatenation of all $\mathbf{h}_k$ is MIL mean/max pooled into this bag's representation vector $\mathbf{z}$, shown in blue. The final prediction $\hat{\theta}(X)$ of the bag's class is depicted in green.

The attention-based embedding-level approach Again, given the LeNet-5 feature extractor, the authors propose to combine the interpretability of the instance-level approach with the advantageous bag-level classification performance of the embedding-level approach. Fig. 6 shows the general idea of the information flow through MIL attention, specifically the weighted average of the embeddings. The attention weights a_k provide

interpretability about key instances inside the bag, which may be responsible for the prediction $\hat{\theta}(X)$ of the bag's class, coming from the bag-level classifier that processes the bag representation $\mathbf{z}$.

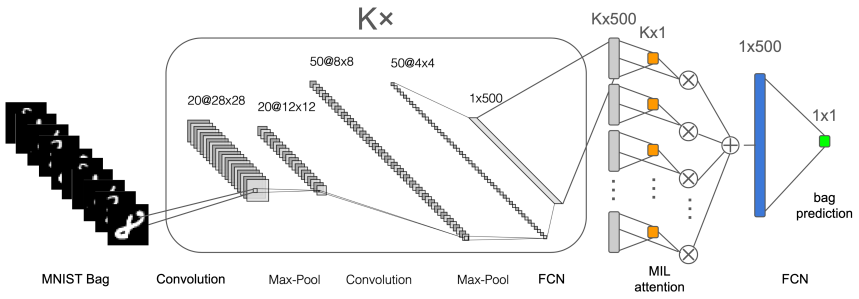

Figure 6: **Attention mechanism embedding model:** For each $\mathbf{x}_k$ in one MNIST bag the feature extractor creates an embedding $\mathbf{h}_k^{1\times d_{\text{feat}}}$. The MIL attention/gated-attention operator is applied to each $\mathbf{h}_k$ and provides the attention weights a_k, depicted in orange. After the weighted average is calculated, the architecture follows the above Fig. 5.

3.3 Optimization objective

A typical objective in Machine Learning is to maximize the likelihood function, which measures how well model parameters explain the observed data. The likelihood function for a single observation follows a Bernoulli distribution and is stated in Eq. 1. Introducing the log-likelihood, the product of probabilities for all observations turns into a sum, which is numerically more stable and better to differentiate for gradient based methods, such as Adam optimization Kingma and Ba (2017). Finally, maximization of log-likelihood is converted into a minimization of neg-log-likelihood, see Eq. 2. For a binary case, the neg-log-likelihood is the same as the well-known binary cross-entropy Bishop (2006). In case of MIL, the objective should measure a discrepancy between true binary labels and predicted probabilities, pushing the model to produce accurate probability estimates for binary outcomes.

$$L(Y, \theta(X)) = p(Y \mid X; \theta) = \theta(X)^Y (1 - \theta(X))^{1-Y} \tag{1}$$

$$-\log L(Y, \theta(X)) = \text{BCE}(Y, \theta(X)) = -(Y \log(\theta(X)) + (1 - Y) \log(1 - \theta(X))) \tag{2}$$

3.4 Validation metrics

As the MIL problem is a classification problem, standard validation metrics Bishop (2006); Mu (2016), namely accuracy/error, precision, recall and area under curve (AUC), serve as good indicators for the performance of the models, as well as detectors of overfitting or numerical instabilities. Moreover, inspection of the attention weights a_k, if accessible, provides meaningful insights into the model's results. The aforementioned metrics are based on: TP (True Positives): $Y = 1$ and $\hat{\theta}(X) = 1$, FN (False Negatives): $Y = 1$ and $\hat{\theta}(X) = 0$, FP (False Positives): $Y = 0$ and $\hat{\theta}(X) = 1$ and TN (True Negatives): $Y = 0$ and $\hat{\theta}(X) = 0$.

Accuracy and Error The accuracy $\text{acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$ is defined as the proportion of correct model predictions out of the total number of predictions. As it is ensured that there are no imbalanced classes, the accuracy serves as a meaningful metric. The error $\text{err} = 1 - \text{acc} = \frac{\text{FP+FN}}{\text{TP+TN+FP+FN}}$ is just the complement of the accuracy.

Precision and Recall $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$ indicates how accurate the model's positive predictions are. A high precision means that most predicted positives are true positives. Additionally, $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$ measures how well the model can identify all true positive cases. A high recall means that most of the actual positives are correctly identified. Recall is also known as the True Positive Rate (TPR). Equivalently, there is the False Positive Rate $(\text{FPR}) = \frac{\text{FP}}{\text{FP+TN}}$.

Area under curve (AUC) The AUC score $= \int_0^1 \mathrm{TPR}(t)\, d(\mathrm{FPR}(t))$ measures the overall performance of a binary classifier by summarizing the Receiver Operating Characteristic curve (ROC), which plots TPR against FPR at various classification thresholds t. AUC represents a probability that a model ranks a randomly chosen positive instance higher than a randomly chosen negative instance. An AUC of 1 indicates perfect performance, where the model ranks all positive instances higher than the negative ones. An AUC of 0.5 suggests no discriminative power, equivalent to random guessing. AUC values less than 0.5 indicate that the model performs worse than random guessing which might be due to dataset issues or a poorly trained model Mu (2016).

3.5 Experiments and Results

In the Milestone 1 experiments, we attempted to reproduce the results of Ilse et al. (2018) for the MNIST-Bags dataset. Therefore, two research questions that needed verification, were: (i) their MIL attention/gated-attention approaches achieve performance better than or comparable to the MIL instance and embedding mean/max approaches, (ii) their approach provides interpretable attention weights, which enable identification of important instances. We refer to the Appendix 6.4. of Ilse et al. (2018) for details on chosen architecture parameters, as well as optimization parameters. For an overview of the specific model approaches, we refer to Fig. 4, 5, 6. Detailed instructions on how to replicate our experiments can be found in the ReadMe.md of our repository. A selection of training logs is provided in this WandB report.

Details In compliance with Ilse et al. (2018), we set up our experiments by varying the mean bag size to 10, 50, and 100, with corresponding variances of 2, 10, and 20. Additionally, we consider different numbers of training bags, namely 50, 100, 150, 200, 300, 400, and 500. The validation is performed on a fixed number of 1000 test bags. The MNIST-Bags dataset is fixedly split into training and test data. For each experiment, five repetitions are carried out and the Adam optimizer Kingma and Ba (2017) with weight decay and a fixed learning rate is used for training. The aforementioned experiment configurations should investigate the effects of varying both the number of training bags and the bag size on the performance of different approaches. We choose to evaluate them using a grid search provided by Weights and Biases WandB (2024). Furthermore, we choose '9' as the key instance determining, whether a bag is given a positive or a negative label, because according to Ilse et al. (2018), it can be easily confused with '4' or '7'.

Training The training is performed using the same configuration as proposed in the paper. Fig. 7 shows the training loss for the embedding approach using attention pooling with the mean bag size of 10 for different numbers of training bags. In general, the neg-log-likelihood converges, and what can be observed is that it happens relatively fast. Yet, for at least three different configurations the training loss unexpectedly increases, and then decreases again to near zero. A potential issue responsible for such behavior is that the learning rate is set too high, which consequently leads to observed instability. Possible solutions include reducing the learning rate or using a learning rate scheduler, which adapts the learning rate over time. Alternatively, the issue could have been caused by gradient explosions, when gradients grow uncontrollably large during backpropagation, destabilizing the learning process. These can be mitigated through techniques such as gradient clipping, or by applying stronger regularization methods, such as adding dropout layers or increasing the 'weight decay' parameter in the Adam optimizer. Nonetheless, the authors included in their optimization strategy a stopping criterion based on the lowest validation error + loss, suggesting that once convergence is reached, potential instabilities are not considered. However, the authors

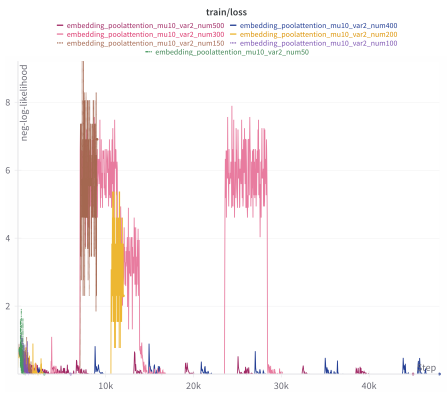

Figure 7: Training loss unexpected behavior

did not specify whether the training should be stopped immediately if there was no improvement in the evaluation metric, or only after several rounds of no improvement. To address this, we have decided to set a patience value of 3, meaning that training will stop, as soon as the validation error + loss does not improve after three consecutive evaluations.

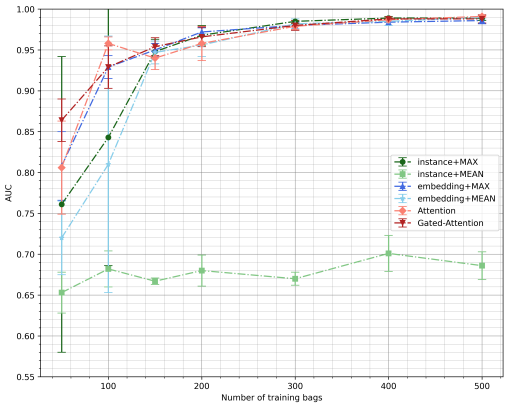

Figure 8: Test AUC with $\mu = 10$

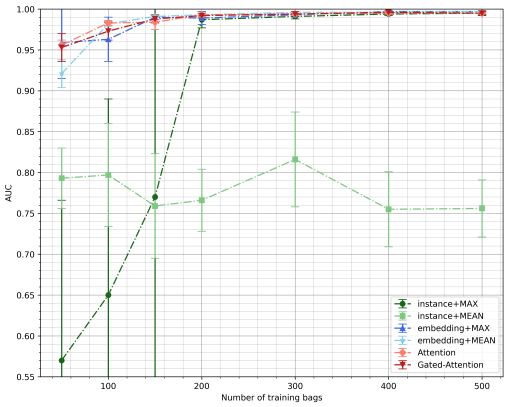

Figure 9: Test AUC with $\mu = 50$

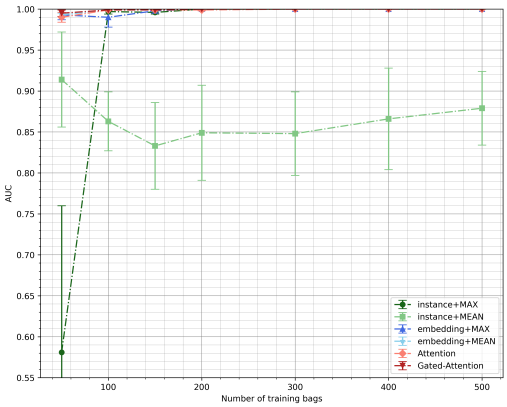

Figure 10: Test AUC with $\mu = 100$

Results The chosen evaluation metric is AUC and in Fig. 8, 9, 10 and Tab. 2, 3, 4, we provide the results for a mean bag size of 10, 50, 100 and corresponding variances of 2, 10 and 20 respectively. The AUC test results from Ilse et al. (2018) have generally been successfully reproduced. Their first research question has been mainly confirmed by our experiments, showing that the attention/gated-attention based MIL approach indeed achieves comparable or better performance compared to existing MIL approaches. In particular, we need to highlight that their statement "(...) the proposed attention-based deep MIL approach performs much better than other methods in the small sample size regime." predominantly holds in our experiments with a mean bag size of 10, see Tab. 2. We want to emphasize that in the current training and evaluation setup, we refrained from further hyperparameter fine-tuning (e.g. learning rate), which could potentially lead to even better results. This matter is further discussed in Sec. 4. We also confirm their findings that the embedding-based models mostly outperform the instance-based models, regardless of mean bag size. Moreover, a sufficiently large number of training bags leads to the observation, that all models, with the exception of the instance approach with mean pooling, produce almost identical results. Furthermore, we confirm that the mean operator performs significantly worse than the max operator, especially coupled with the instance-based approach. Lastly, their second research question has also been confirmed by our experiments. The attention mechanism provides interpretability by highlighting key instances of a bag, as those having higher attention weights a_k. The meaningful attention weights for a positive bag, where '9' is the key instance, are presented in Fig. 11. Upon inspection of ambiguously looking instances, for which it is unclear which digit they represent, we identify the limits in interpretability of the attention mechanism due to insufficient data quality. In Fig. 12, we show such ambiguous instances ('9' $\rightarrow$ '4', '7') in the MNIST-Bag, which leads to smaller a_k in those cases. Fig. 13 shows the almost uniformly distributed a_k for a negative MNIST-Bag.

4 Discussion

4.1 Challenges & Possibilities

The MNIST dataset is a widely used benchmark dataset to try out different concepts, prototypes and baseline models, which provides a large number of samples. The dataset creation for the MIL problem is rather straightforward, as long as one takes into account that a number of instances inside a bag should vary and that the MNIST-Bags dataset needs to be balanced, especially when the mean bag size is 50 or 100. Without such balance, there is a high probability of including a positive instance in an MNIST-Bag, given the limited diversity of instance labels (only ten digits), as mentioned in Sec. 2.2. To ensure interoperability between embedding-based and instance-based approaches, it is essential to carefully curate each batch, comprising the bag itself, the bag label, and individual instance labels.

The concept of providing only a bag-level label to the model, while enabling it to identify key instances within the bag through an attention mechanism, is inherently complex. Yet, such an approach allows the model to determine the instances responsible for classifying the bag as positive or negative. Qualitative results from the MNIST-Bags dataset have clearly validated effectiveness of this approach, see A.2.

The upcoming experiments using a real-world histopathology dataset will be intriguing not only in terms of general model performance, but also explainability of the model. The structured nature of these datasets could facilitate data curation, and the experiments will show whether MIL approaches can effectively handle high-resolution images, divided into smaller patches, while making computation more manageable. In histopathology, slide-level labels (e.g. "cancerous") are commonly available, as annotating at the instance/patch level (e.g. for individual cells) is highly labor-intensive. MIL methods enable models to learn from these weak, slide-level labels, aligning well with clinical workflows where detailed annotations are often impractical. This means from a business perspective that a well-trained medical professional does not have to spend time on the detailed annotation of datasets, but can carry out such data processing in a shorter time and therefore more cost-effectively. For practical implementation, the models will be trained using curated bag-level labels rather than raw slide-level labels, just as they were in case of MNIST-Bags within the scope of milestone 1.

4.2 Future Work

Overall, no significant issues were encountered during milestone 1. A considerable amount of time was dedicated to establishing a modular codebase, ensuring it is easily adjustable for upcoming experiments involving the real-world histopathology dataset. The detailed appendix in Ilse et al. (2018), where the authors specified the hyperparameters for baseline models and their approach, facilitated the successful replication of the baseline experiments. In the current setup, we refrained from fine-tuning hyperparameters, such as learning rate or weight decay, instead using the values proposed in the original paper. The results of our MNIST-Bags experiments are consistent with those reported in Ilse et al. (2018).

For milestone 2, we propose conducting a thorough hyperparameter fine-tuning. Additionally, given the sparsity of medical imaging datasets, we may employ N-fold cross-validation to robustly evaluate model performance. Our main objective is to adapt the existing training pipeline to the histopathology dataset and replicate the corresponding results reported in the paper.

In future research, we aim to delve deeper into the internal structure of the model architectures. This might uncover interpretable features, neurons, or layers beyond the attention weights a_k that could provide further insights into the success of MIL.

References

D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate, May 2016. URL http://arxiv.org/abs/1409.0473. arXiv:1409.0473.

C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language Modeling with Gated Convolutional Networks, Sept. 2017. URL http://arxiv.org/abs/1612.08083. arXiv:1612.08083.

T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(96)00034-3. URL https://www.sciencedirect.com/science/article/pii/S0004370296000343.

J. Feng and Z.-H. Zhou. Deep MIML Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v31i1.10890. URL https://ojs.aaai.org/index.php/AAAI/article/view/10890.

M. Ilse, J. Tomczak, and M. Welling. Attention-based deep multiple instance learning. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2127–2136. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ilse18a.html.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, Jan. 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov. 1998. ISSN 1558-2256. doi: 10.1109/5.726791. URL https://ieeexplore.ieee.org/document/726791/?arnumber=726791. Conference Name: Proceedings of the IEEE.

Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A Structured Self-attentive Sentence Embedding, Mar. 2017. URL http://arxiv.org/abs/1703.03130. arXiv:1703.03130.

A. C. Mu. Introduction to Machine Learning with Python. 2016.

P. O. Pinheiro and R. Collobert. From Image-level to Pixel-level Labeling with Convolutional Networks, Apr. 2015. URL http://arxiv.org/abs/1411.6228. arXiv:1411.6228 [cs].

C. Raffel and D. P. W. Ellis. Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems, Sept. 2016. URL http://arxiv.org/abs/1512.08756. arXiv:1512.08756.

WandB. Weights & Biases: The AI Developer Platform — wandb.ai. https://wandb.ai/site/, 2024. [Accessed 13-11-2024].

X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu. Revisiting Multiple Instance Neural Networks, Oct. 2016. URL http://arxiv.org/abs/1610.02501. arXiv:1610.02501.

K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, Apr. 2016. URL http://arxiv.org/abs/1502.03044. arXiv:1502.03044.

W. Zhu, Q. Lou, Y. S. Vang, and X. Xie. Deep Multi-instance Networks with Sparse Label Assignment for Whole Mammogram Classification, May 2017. URL http://arxiv.org/abs/1705.08550. arXiv:1705.08550.

A Appendix

A.1 Quantitative Results

The test AUC values for 10, 50, and 100 instances per MNIST-Bag, on average, are reported for a varying number of training MNIST-Bags in Tables 2, 3, and 4.

Table 2: Test AUC, $\mu = 10$, for MNIST-Bags dataset with a different number of training bags.

# train bags	50	100	150	200	300	400	500
Instance+max	0.761 ± 0.181	0.843 ± 0.157	0.948 ± 0.015	0.968 ± 0.012	**0.985 ± 0.003**	**0.989 ± 0.002**	0.989 ± 0.003
Instance+mean	0.653 ± 0.025	0.682 ± 0.022	0.667 ± 0.004	0.680 ± 0.019	0.670 ± 0.008	0.701 ± 0.022	0.686 ± 0.017
Embedding+max	0.808 ± 0.042	0.929 ± 0.014	0.950 ± 0.008	**0.972 ± 0.005**	0.980 ± 0.004	0.984 ± 0.004	0.986 ± 0.004
Embedding+mean	0.720 ± 0.045	0.810 ± 0.157	0.947 ± 0.014	0.956 ± 0.014	0.982 ± 0.006	0.985 ± 0.005	0.990 ± 0.002
Attention	0.806 ± 0.057	**0.958 ± 0.008**	0.948 ± 0.014	0.958 ± 0.021	0.979 ± 0.004	0.987 ± 0.003	**0.991 ± 0.001**
Gated Attention	**0.864 ± 0.026**	0.929 ± 0.026	**0.954 ± 0.011**	0.966 ± 0.012	0.980 ± 0.006	0.988 ± 0.001	0.988 ± 0.005

Table 3: Test AUC, $\mu = 50$, for MNIST bags dataset with a different number of training bags.

# train bags	50	100	150	200	300	400	500
Instance+max	0.570 ± 0.196	0.650 ± 0.240	0.770 ± 0.264	0.987 ± 0.010	0.991 ± 0.003	0.994 ± 0.002	0.995 ± 0.002
Instance+mean	0.793 ± 0.037	0.797 ± 0.063	0.759 ± 0.064	0.766 ± 0.038	0.816 ± 0.058	0.755 ± 0.046	0.756 ± 0.035
Embedding+max	**0.959 ± 0.044**	0.963 ± 0.027	0.990 ± 0.003	0.989 ± 0.008	0.993 ± 0.002	**0.997 ± 0.001**	**0.997 ± 0.001**
Embedding+mean	0.921 ± 0.017	0.982 ± 0.003	**0.991 ± 0.001**	**0.993 ± 0.002**	**0.996 ± 0.001**	0.996 ± 0.002	**0.997 ± 0.001**
Attention	0.956 ± 0.006	**0.983 ± 0.003**	0.984 ± 0.009	**0.993 ± 0.002**	0.994 ± 0.001	0.995 ± 0.002	0.995 ± 0.001
Gated Attention	0.953 ± 0.017	0.973 ± 0.011	0.988 ± 0.005	0.992 ± 0.003	0.994 ± 0.003	0.996 ± 0.002	0.995 ± 0.003

Table 4: Test AUC, $\mu = 100$, for MNIST bags dataset with a different number of training bags.

# train bags	50	100	150	200	300	400	500
Instance+max	0.581 ± 0.179	0.997 ± 0.003	0.996 ± 0.002	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
Instance+mean	0.914 ± 0.058	0.863 ± 0.036	0.833 ± 0.053	0.849 ± 0.058	0.848 ± 0.051	0.866 ± 0.062	0.879 ± 0.045
Embedding+max	0.993 ± 0.006	0.990 ± 0.012	0.998 ± 0.003	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
Embedding+mean	0.993 ± 0.002	**0.999 ± 0.000**	**0.999 ± 0.001**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
Attention	0.990 ± 0.006	0.999 ± 0.001	**0.999 ± 0.001**	0.999 ± 0.001	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
Gated Attention	**0.995 ± 0.004**	0.999 ± 0.001	**0.999 ± 0.001**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**

A.2 Qualitative Results

The attention weights a_k for different MNIST-Bags with 10 instances per bag on average. Figure content is best viewed when zoomed in.

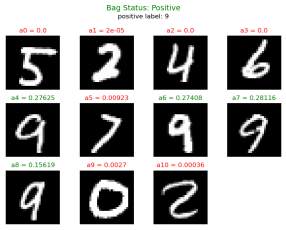

Figure 11: Attention weights a_k for a positive bag, $\mu = 10$.

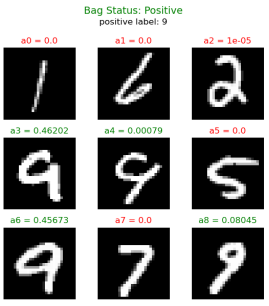

Figure 12: Ambiguous instances and corresponding a_k.

Figure 13: Attention weights a_k for a negative bag, $\mu = 10$.