Zero-shot Learning using DAP and ESZSL

Deep Learning and Spiking Neural Networks: Final project

Adam Cihlar[1]

27. June 2023

Abstract: This paper explores zero-shot learning on the AWA2 and CUB datasets, developing and evaluating multiple models. The best-performing approach, An embarrassingly simple approach to zero-shot learning (ESZSL), which provides an analytical solution of the optimization problem and makes the training very efficient, outperforms other methods based on direct attribute prediction approach. The incorporation of BERT semantic embeddings as an enrichment of the attribute information did not improve the performance. The ESZSL model achieves an accuracy of 52.8 % on AWA2's unseen class test set and 53.9 % on CUB's test set. For test set consisting of classes seen during training, accuracies were 92.4 % (AWA2) and 64.1 % (CUB).

Keywords: Zero-shot learning, DAP, ESZSL, AWA2, CUB

Introduction

In traditional machine learning, the classification relies on predefined classes and labelled data. Zero-shot image classification represents a paradigm shift in the field, as it offers a new approach to classify instances without the need for explicit training data, unlike traditional methods that rely on exhaustive labeled examples for each class. In zero-shot classification we leverage auxiliary information and semantic relationships to find the relationships between seen and unseen categories. In this work we develop and compare several approaches to zero-shot image classification.

The structure of the paper goes as follows. First, we introduce the data, their structure and meaning of single components and describe various preprocessing techniques that we will apply. Then, we introduce the selected models and training procedures that we utilize to optimize the single models, and depict the whole flow of the solution in a diagram. For each model, we experiment with several modifications and choose the best performing one. In chapter 3, we report the results of the selected models. Finally, we elaborate on the achieved performance.

[1]University of Klagenfurt, Faculty of Technical Sciences, Department of Smart Systems Technologies, Degree: Information and Communications Engineering, adcihlar@edu.aau.at

1 Data and preprocessing

Throughout the work, we will utilize two zero-shot learning datasets, that are well recognized and used as a benchmark for newly presented models. Both datasets follow the same structure that we will introduce in the next subsection.

1.1 AWA2

First dataset is Animals with Attributes 2 introduced by Xian et al. (2017). The dataset consists of $37,322$ images of 50 animals classes with pre-extracted feature representations for each image. The features serve to isolate the effect of zero-shot learning algorithms and to avoid boosting the performance of the zero-shot model by using a stronger feature extractor, therefore the zero-shot learning models are comparable. The features were extracted by pretrained ResNet model (He et al., 2016), that was pre-trained on ImageNet with 1K classes. The image feature vectors are of size 2,048. For each class, we also have an attribute vector of length 85 describing the properties of the animal. The attribute vectors are available in various forms, binary, continuous and continuous in the interval of $[0, 1]$. Based on the selected model, we will choose a suitable representation of the attributes, the discussion will follow in section 2.

For reproducible results, the authors propose a predefined splits, these splits were later refined by Xian et al. (2018) and the latter version became a standard in the machine learning community. The overview of the data and splits are presented together with the second dataset in table 1.

Figure 1: Sample from the AWA2 dataset

1.2 CUB

Second dataset at hand is the Caltech-UCSD Birds-200-2011 dataset (CUB) (Wah et al., 2011) is specifically designed for fine-grained bird species recognition. It contains images of 200 bird species, with a total of $11,788$ images. The dataset covers a wide range of bird species, including both common and rare species, and provides a challenging task for classification algorithms due to the subtle visual differences between closely related species.

Same as the AWA2 dataset, it comes with features pre-extracted from the images by ResNet model, thus again having dimensionality of 2,048. The class attribute vectors consists of 312 features and also come in the three forms - binary, continuous and continuous in the interval of

$[0, 1]$. The dataset is also divided to predefined splits (see table 1) as suggested by Xian et al. (2018).

Figure 2: Sample from the CUB set

1.3 Splits and preprocessing

To ensure equal conditions and continuity for developing the zero-shot learning models, we stick to the predefined data splits from Xian et al. (2018). Importantly, the splits also define seen and unseen classes during training. For AWA2 dataset from the total count of 50 classes, 10 classes are held-out and never seen by the model during training, for CUB dataset it is 50 out of 200 classes. For training the models, we further utilize predefined split of the training dataset to training and validation sets, which will allow us for performing hyper-parameter tuning.

The preprocessing procedures, such as choosing the attribute representations, reducing the visual features dimensionality or extending the attributes with semantic text encodings, will be applied model-specifically. Therefore, we will introduce the techniques later in section 2 with the corresponding models.

		Number of classes			Number of images				
						At training time		At testing time	
Dataset	Attributes	Y_{total}	Y_s	Y_u	Total	Y_s.	Y_u	Y_s.	Y_u
AWA2	85	50	40	10	37,322	23,527	0	5,882	7,913
CUB	312	200	150	50	11,788	7,057	0	1,764	2,967

Table 1: Pre-defined dataset splits

Note. Y_s stands for seen classes or instances of seen classes for the right part of the table, while Y_u stands for the classes never seen by the models during training, therefore number of images for training time with label $Y_u \in Y_u$ is always zero.

2 Models and training procedures

In general, we utilize two different approaches. First, following the concept of Direct Attribute Prediction (Lampert et al., 2013), and second, we apply An embarrassingly simple approach to zero-shot learning introduced by Romera-Paredes et al. (2015).

2.1 Direct Attribute Prediction

Direct Attribute Prediction (DAP) overcomes the problem of unavailable labelled data of the test set by predicting the single attributes instead of the label and inferring the class based on the predicted attributes of a sample (Lampert et al., 2013). Using the available labelled data, we build a single classifier or regressor for each attribute, based on the chosen attribute representation. In the original paper, the authors use Support Vector Machines as the underlying model. In our case, we experiment with a set of multilayer perceptrons (MLP).

2.1.1 Multi-model classification

For the AWA2 dataset we use the binary representations of attributes and train 85 MLP classifiers. To reduce the size of the models, we apply principal component analysis and transform the original feature vectors from 2,048 dimensions to 512. Applying this transformation, we reduce the size of the models almost 4 times and at the same time preserve 91.2 % of the variance of the original data. The size of the input layer thus becomes 512, hidden layer consists of 256 neurons and the output of the model is a probability of the image containing the attribute or not. Therefore, we apply sigmoid activation function on top of the final layer. The hidden layer is followed be ReLU activation function. To avoid over-fitting, we apply dropout equal to 0.2.

We optimize the models for 2 epochs using Adam optimizer (Kingma et al., 2014) in its original settings with learning rate 0.001 and minimize the binary cross-entropy loss function for each classifier.

$$BCE = -\sum_{i}^{N} (y^i \log(pred^i) + (1 - y^i) \log(1 - pred^i)) \tag{2.1}$$

During inference, we predict probability of each attribute for the given test sample using the trained 85 models, thus we obtain a vector of probabilities of length 85. After that, we search for the nearest class in the attribute space using Euclidean distance between the one-hot encoded ground truth attributes of the classes and the predicted probabilities. We also experiment with thresholding the probabilities and finding the closest class using Hamming distance.

As the CUB dataset contains 312 attributes, this approach would be inconvenient, not only for the training, but also for inference, as we would need to get predictions from 312 different models. Therefore, we will not consider using this approach for CUB dataset.

2.1.2 Joint-model classification

To mitigate the obstacle of training a single model for each attribute prediction, we will model the attributes as a distribution and train a single large model that will predict the vector of

probabilities of all attributes at once.

As we train only 1 model now, we do not reduce the dimensionality by PCA and work with the fully sized feature vectors. Therefore, the input layer is of size 2048, we also extend the size of the hidden layer to 1024 neurons, the size of the last layer corresponds to the nubmer of attributes of a dataset, thus being 85 for AWA2 dataset and 312 for CUB dataset, respectively. The hidden layer is followed by ReLU activation function and we apply sigmoid activation function for every neuron in the output layer as the attribute prediction can be treated as multi-label classification.

We train the models by minimizing the sum of binary cross-entropy over all output neurons and optimize the parameters using Adam optimizer. For AWA2 dataset we use learning rate 1e-4 and train the model on training data for 20 epochs and evaluate the performance after each epoch on validation set. Then, we retrain the model on both training and validation data using the optimal number of epochs. We utilize the same procedure for the CUB dataset, but with learning rate of 2.5e-5 and maximum of 300 epochs. The optimal number of epochs for AWA2 is 5 and 296 for CUB, respectively.

$$BCE_{multilabel} = -\sum_{i}^{N}\sum_{c}^{C}(y_c^i \log(pred_c^i) + (1 - y_c^i)\log(1 - pred_c^i)) \tag{2.2}$$

2.2 An embarrassingly simple approach to zero-shot learning

An embarrassingly simple approach to zero-shot learning (ESZL) was introduced in 2015 by Romera-Paredes et al. and despite its simplicity outperformed the existing, more sophisticated approaches. The approach relies on learning a straightforward linear mapping from the visual feature space to the attribute space while imposing regularization on the learned weight matrix. Formally, that can in general be expressed by:

$$\text{minimise}\, L(X^T V\, S,\, Y) \,+\, \Omega(V), \tag{2.3}$$

where L stands for the chosen loss function, X stands for the visual feature matrix, V for the matrix of learnable parameters, S for the matrix of attributes and Y for the vector of labels. The Ω is a regularization term. With the loss function being the Euclidean distance and a certain choice of the regularization term, the minimization is convex and its global optimal solution can be computed efficiently using the following formula:

$$V \,=\, (XX^T \,+\, \alpha I)^{-1}\, XYS^T\, (SS^T \,+\, \gamma I)^{-1}, \tag{2.4}$$

where α and γ are the regularization hyper-parameters. We use the training and validation splits to search for the best combination of these hyper-parameters for both datasets. We consider the hyper-parameters ranges for α and γ as suggested in the original paper, i.e. 10^b for b $= -3, -2, \ldots, 2, 3$. The authors argue that because of the Euclidean distance, the best choice of the attributes is their continuous representation in the interval $[0, 1]$. To estimate the ESZSL model we use the implementation of Sbharadwajj (2018).

In the second step, we experiment with enriching the attributes with semantic embeddings of the labels of individual classes. We utilize a pretrained BERT model (Devlin et al., 2018) to embed the labels, we pass the whole name of the label and extract the CLS token embeddings

from last layer of the encoder to serve as a semantic representation of the label. However, as the embedding is a vector of length 768, we utilize PCA to extract only the most differentiating features of each class label. Namely, we reduce the dimensionality to 10 while preserving 57.7 % of the variability of the AWA2 semantic label embeddings and 42 % in the case of the CUB dataset, respectively. To conform to the expected range of values, we normalize the obtained vectors to values of interval $[0,\ 1]$ using minmax normalization. Finally, we concatenate these dense vectors to the original attribute vectors obtaining new attribute vectors of size 95 (AWA2) and 322 (CUB), and estimate the ESZSL model on these new attribute vectors. We will denote this extension as ESZSL+.

We evaluate all the models, Multi-MLP Classifier, Joint-MLP Classifier, ESZSL and ESZSL+, on both test datasets and compute accuracy and weighted F1 score for both. Finally, we combine the metrics obtained from the single datasets and report their harmonic mean.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

$$F1_{weighted} = \sum_{c}^{C} \frac{|N_c|}{|N|} \frac{2 * Precision_c * Recall_c}{Precision_c + Recall_c} = \sum_{c}^{C} \frac{|N_c|}{|N|} \frac{2 * TP_c}{2 * TP_c + FP_c + FN_c} \tag{2.6}$$

$$\text{Harmonic mean (metric)} = \frac{2 \times metric_s \times metric_u}{metric_s + metric_u} \tag{2.7}$$

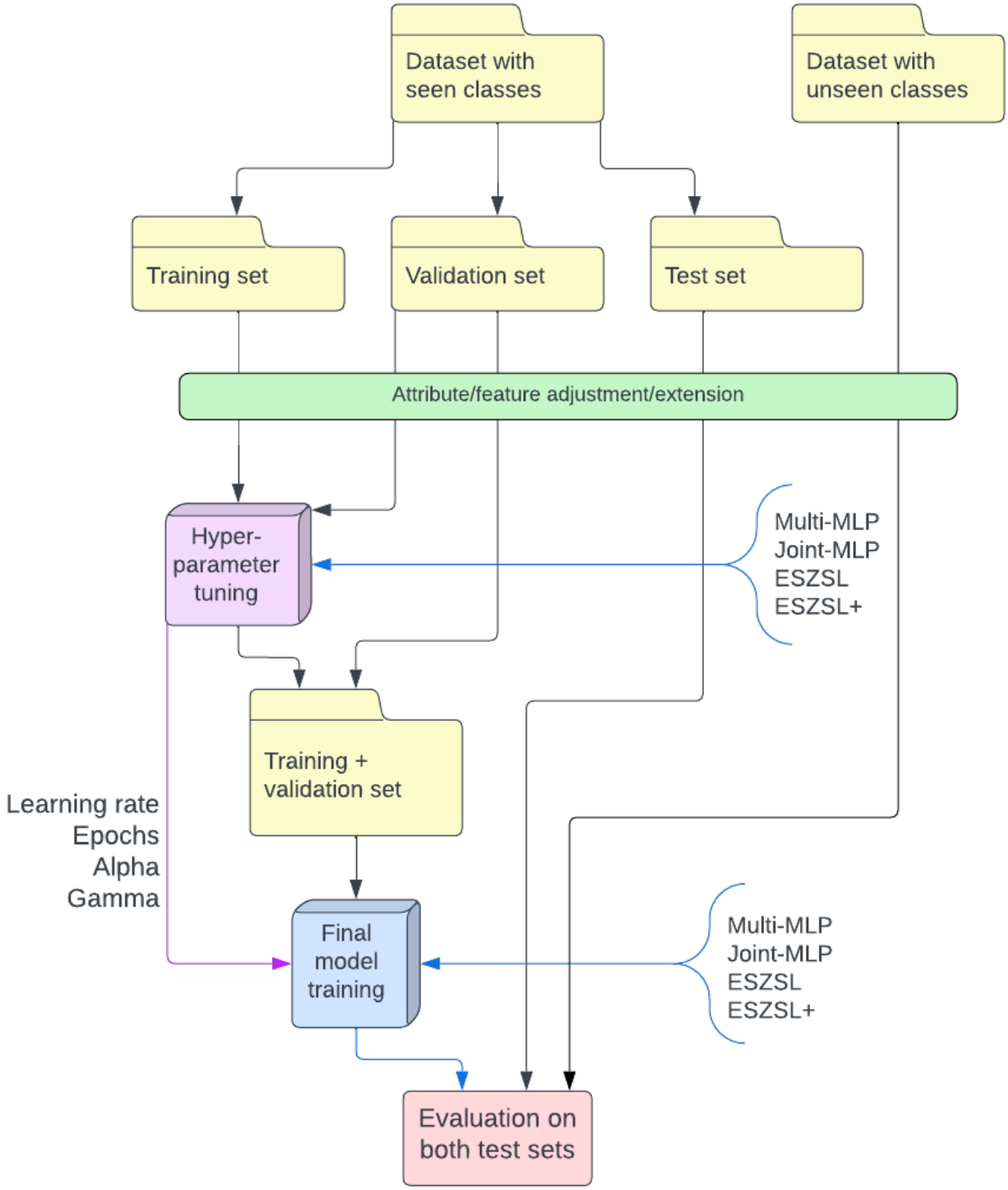

Figure 3: Diagram of the pipeline

Note. Yellow components represent data, green components represent attribute or features preprocessing, such as dimensionality reduction of the visual features or binarizing of the attributes or extending them with BERT embeddings. The purple box depicts hyper-parameter tuning stages, the blue one final training of a model and red box stands for evaluation of a trained model. Black arrows depict the flow of data, purple arrow the flow of optimal hyper-parameters and blue arrows stand for flow of the models.

3 Evaluation

In this section, we will present how the four developed models compare to each other on test datasets consisting of seen classes and its counterparts with unseen classes. We evaluate the performance using accuracy, weighted F1-score for each dataset separately and also report the harmonic mean of the two datasets of the individual metrics.

		Unseen classes		Seen classes		Harmonic mean	
Model	Distance metric	Acc.	F1-score.	Acc.	F1-score	Acc.	F1-score
Multi-MLP	Euclidean	36.1	31.5	93.8	93.8	52.1	47.2
Multi-MLP	Hamming	33.1	29.1	93.8	93.8	48.9	44.4
Joint-MLP	Euclidean	33.2	27.7	**94.5**	**94.4**	49.1	42.8
Joint-MLP	Hamming	32.6	26.9	94.3	94.3	48.5	41.9
ESZSL	**Euclidean**	**52.8**	**52.0**	92.4	92.1	**67.2**	**66.5**
ESZSL+	Euclidean	34.7	36.9	91.1	91	50.3	52.5
*DAP	Hamming	46.1	-	-	-	-	-
*ESZSL	Euclidean	58.6	-	-	-	-	-

Table 2: Comparison of models on AWA2 test sets

*Note. The utilized methods were introduced before creation of the datasets, thus we cannot find the metrics for the datasets in the original papers. However Xian et al. (2018) trained and evaluated the models on the newly proposed datasets. We report their results in the bottom of the table and denote them by *.*

On the AWA2 dataset, we can compare the two approaches of direct attribute prediction, the multi and joint model approach. The first one outperforms the latter on test set with unseen classes, while the second is slightly superior to the first if compared by performance on test set with seen classes. If we evaluate the models by the harmonic means of the metrics, the multi-MLP model, where we have a separate classifier for each attribute, achieves higher metrics by approximately 3 to 4 points. Using Euclidean metric instead of Hamming for inferring the nearest class proved to perform better in all evaluated settings.

However, despite its simplicity, ESZSL is clearly the best performing model of all on the unseen classes and loses only by a small margin on the seen classes, which makes it unambiguously the best model overall. After enriching the attributes with semantic embeddings of the class labels, the performance surprisingly drops to the level of multi-MLP model. We assume that the embeddings does not contain enough variance as we are operating in very specific domain of animals, while the BERT model was pretrained on texts of all kinds and domains.

Xian et al. (2018) retrained and evaluated the proposed methods on both datasets. Unfortunately, they report only the accuracy on test set consisting of unseen classes, thus the comparison of our models can not be comprehensive. However, based on the available metric, the DAP model outperforms all our DAP-based models and the same applies for ESZSL. We assume that the success of using SVM as the underlying classifier for DAP might be based on its better generalization properties compared to multi-layer perceptron, which might be prone to overfitting. This idea is supported by superior results of the Joint-MLP model on the test set consisting of

only seen classes. We assume that the better results of ESZSL from Xian et al. (2018) might have been reached by finer hyper-parameter tuning.

		Unseen classes		Seen classes		Harmonic mean	
Model	Distance metric	Acc.	Acc.	F1-score	F1-score	Acc.	F1-score
Multi-MLP	Euclidean	-	-	-	-	-	-
Multi-MLP	Hamming	-	-	-	-	-	-
Joint-MLP	Euclidean	19.8	15.2	**72.8**	**72.5**	31.1	25.1
Joint-MLP	Hamming	18.2	13.8	72.4	72.1	29.1	23.2
ESZSL	**Euclidean**	**53.9**	**50.8**	64.1	62	**59.1**	**55.8**
ESZSL+	Euclidean	50.8	47.2	65.1	63.4	57.1	54.1
*DAP	Hamming	40.0	-	-	-	-	-
*ESZSL	Euclidean	51.9	-	-	-	-	-

Table 3: Comparison of models on CUB test sets

*Note. The utilized methods were introduced before creation of the datasets, thus we cannot find the metrics for the datasets in the original papers. However Xian et al. (2018) trained and evaluated the models on the newly proposed datasets. We report their results in the bottom of the table and denote them by *.*

The results on the CUB dataset are in line with those on AWA2 dataset. The best performance overall and on unseen classes is again reached by plain ESZSL, while the Joint-MLP model using Euclidean metric achieves the best results on the seen classes test set. Unlike in the previous case, concatenating the attributes with semantic representation of the labels did not lead to a drastic decline in the performance and on the test set with seen classes the metrics were even slightly improved. However, the best overall performance is still reached by the unadjusted ESZSL model.

Compared to the results of Xian et al. (2018), our ESZSL model reached the best score overall, while using the SVM model for the direct attribute prediction proved to be clearly superior to the multi-layer perceptron on the unseen classes test set.

To elaborate on the performance of the best model, ESZSL, we plot confusion matrices of its predictions on the test set with unseen classes for both datasets (figure 4 and 5). To get the representation in percentage we divided the individual elements by the column-wise sum, so the diagonal of the matrix represents the precision for individual classes. We can see that if the model is erroneous, the mispredicted label comes in most of the cases from the same family of animals as the ground truth label. For example, if the model predicts label seal, it is correct with probability of 58 % and in 24 % of the cases the ground truth label is seal, which we can consider as a reasonable mistake, as the two classes share a lot of attributes. Even though the number of classes does not allow for displaying the individual percentages, we can still observe that the errors of the models are more uniformly distributed across the classes, as the CUB dataset consists only of species of birds.

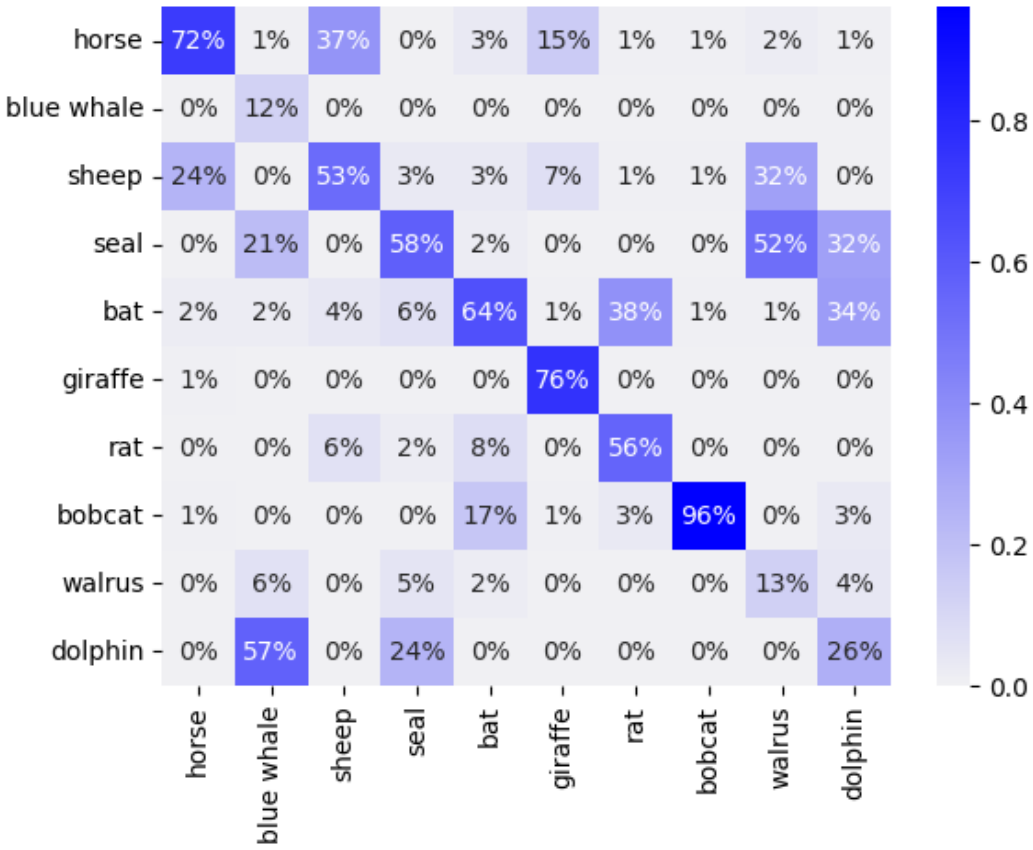

Figure 4: Confusion matrix of predictions on AWA2 dataset

Note. To get the representation of the confusion matrix in percentage we divided the individual elements by the column-wise sum, so the diagonal of the matrix represents the precision for individual classes.

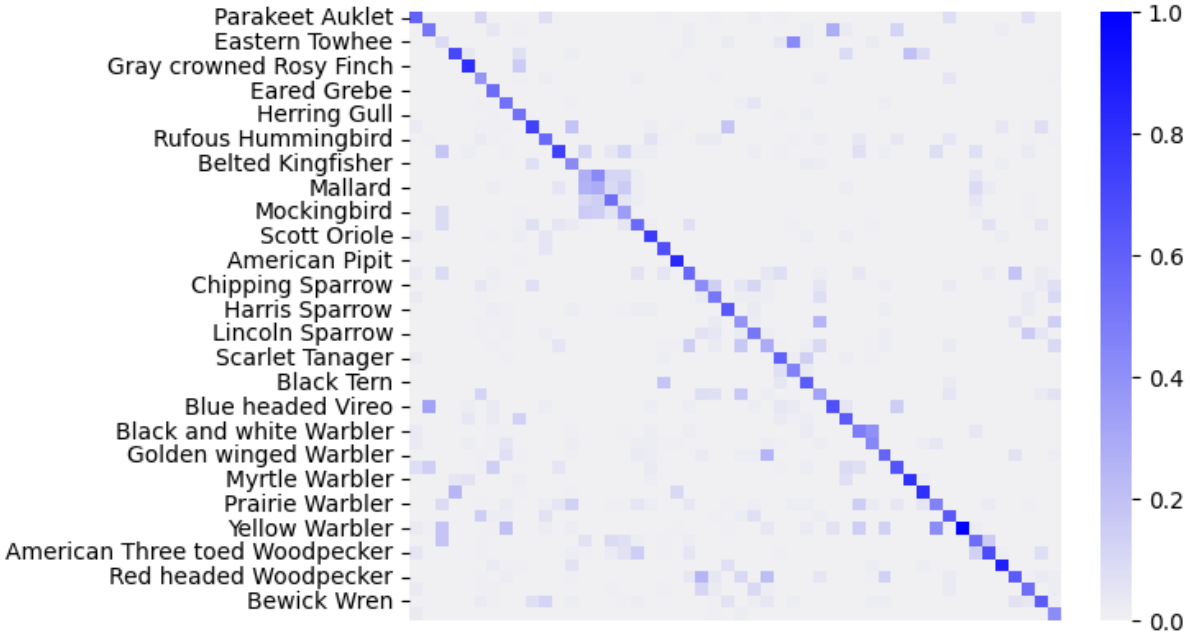

Figure 5: Confusion matrix of predictions on CUB dataset

Conclusion

In this work we developed and evaluated four models for zero-shot learning on AWA2 dataset and three models for CUB dataset. First, we familiarized ourselves with the structure of the data and chose the correct splits of data for the selected approaches. The second chapter was dedicated to description of the individual models. For AWA2 dataset we experimented with two versions of direct attribute prediction approach. The first, relying on predictions of individual attribute classifiers, the second based on one joint model for prediction of all attributes at once. As of the attribute dimensionality of the CUB dataset, we implemented only the latter for that case. The second chosen approach was An embarrassingly simple approach to zero-shot learning (ESZSL) which learns a linear mapping from the feature space to the attribute space while constraining the learned parameters by a regularization term. Despite its simplicity, this approach outperformed all the other considered methods. Finally, we enriched the attribute information with semantic embeddings of the class labels and reestimated the ESZSL model. However, this extension did not bring any performance boost. In last chapter, we presented the results on both types of test sets and compared our results to implementations of authors of our two datasets. Finally, we analyzed the errors of the best-performing models with confusion matrix.

Using the ESZSL model, we reached accuracy of 52.8 % and on the AWA2 test dataset with unseen classes and 53.9 % on the CUB test dataset. The accuracy on the test datasets with classes seen during the training was 92.4 % for AWA2 dataset and 64.1 for the CUB dataset, respectively.

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[3] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[4] Lampert, C. H., Nickisch, H., & Harmeling, S. (2013). Attribute-based classification for zero-shot visual object categorization. IEEE transactions on pattern analysis and machine intelligence, 36(3), 453-465.

[5] Romera-Paredes, B., & Torr, P. (2015, June). An embarrassingly simple approach to zero-shot learning. In International conference on machine learning (pp. 2152-2161). PMLR.

[6] Sbharadwajj. (2018). Embarrassingly-simple-zero-shot-learning. GitHub. https://github.com/chichilicious/embarrsingly-simple-zero-shot-learning

[7] Xian, Y., Lampert, C. H., Schiele, B., & Akata, Z. (2018). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence, 41(9), 2251-2265.

[8] Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset.