Raw Parsed


```
{
  "query": "deep learning",
  "num_results": 3,
  "results": [
    {
      "chunk": "stable isomers. The analogy to machine learning loss landscapes (ML-LLs) is straightforward, the main difference perhaps being that non-minima are valid configurations for sets of weights. Due to this similarity between ELs and ML-LLs, various, well-established methods from the field of energy landscapes can be employed to study ML-LLs. One key area of interest here is interpretability. Employing well-understood methods from a mature field, with a solid mathematical basis in the physical world, to move away from black-box machine learning models may be a helpful step towards interpretable machine learning models. 1 arXiv:2304.02381v2 [cs.LG] 15 Dec 2024 Accepted at the ICLR 2023 Workshop on Physics for Machine Learning 1.2 RELATED WORK Various approaches to interpretability in deep learning for neural networks exist. Below, we are mostly interested in gradient-based methods due to their applicability to non-image data. Vari- ous other methods to interpret the output of CNNs on images exist, as for example summarised in Linardatos et al. (2020), but will not be reviewed below. Gradient-based methods: All gradient-based methods are concerned with changes in the predic- tion as the input data is slightly perturbed. For a vector-valued input x ⊂X ∈Rd and some loss function, L, a gradient-based method computes some expression of the form ∂L/∂x, usually for each input node individually. Gradient-based methods were first introduced for images by Si- monyan et al. (2013), who used them to compute how changes in the input affect predictions in the neighbourhood of the input, allowing the computation of a salience map (K¨ummerer et al., 2014; Zhao et al., 2015). More recently, integrated gradient methods (Sundararajan et al., 2017) consider the derivative of the output (loss) with respect to individual input nodes. If the change in loss is large with respect to some input feature, that feature is more likely to be relevant to the decision making. Various other gradient and perturbation based methods exist (Alvarez-Melis & Jaakkola, 2018), yet their usefulness and accuracy is debated, and is generally agreed to be insufficient (Srini- vas & Fleuret, 2020). Energy landscapes in machine learning: Energy landscapes methods have been employed to study machine learning in previous contributions (Ballard et al., 2017; Chitturi et al., 2020). Niroomand et al. (2022) used energy landscapes to characterise new loss functions, and the landscapes view has been used more broadly to gain insights into machine learning models (Segura et al., 2022; Verpoort et al., 2020). Lastly, other applications of energy landscape methods have employed various con- cepts from physical sciences in machine learning, including the heat capacity (Bradley et al., 2022; Niroomand et al., 2022), both for characterisation and model improvement. Interpreting energy landscapes: Due to the associated physical meaning, energy landscapes are usually more easily interpretable. Only minima represent equilibrium configurations, and each min- imum is associated with a unique structure. However, for larger, complex molecules, many minima may exist, and enumerating them may be infeasible. Instead, common features between sets of min- ima, grouped by their energetic properties, may be identified. For example, in (R¨oder et al., 2020) and (R¨oder & Wales, 2022) a multi-funnelled landscape",
      "metadata": {
        "paper_id": "2304.02381v2",
        "chunk_index": 1,
        "total_chunks": 7
      },
      "distance": 1.0201900005340576,
```

```
    "rank": 1
  },
  {
    "chunk": "intervals. Additionally, to prevent the release of ML methods that appear sophisticated but perform no better than simpler algorithms, baselines should be compared to the ‘sophisticated’ method and proven to be statistically inferior (e.g. small vs. deep neural networks). Open areas and limitations of the proposed recommendations The field of Machine Learning in biology is vast. As such, it would be a herculean task to identify cross-cutting recommendations that could be applicable to all branches of ML across all possible domains. Therefore, this work focuses exclusively on supervised ML applications, excluding unsupervised ML, clinical applications, and individual implementations through pipelines to name a few. Moreover, the primary goal of this work is to define requirements and best practices that can be of use in the reviewing process and creation of ML-related papers, while remaining agnostic as to the actual 8 underlying solutions. The intent is to trigger a discussion in the wider ML community leading to future work addressing possible solutions. Several key issues related to reproducibility (e.g. data is not published, data splits are not reported and model source code with its final parameters and hyperparameters are not released) can be aided by a multitude of workflow systems that help to ensure and automate multi-step processes are completely reproducible by tracking model parameters and exact versions of the source code and libraries. Examples of commonly used workflows include Galaxy36 and Nextflow37 . Another de facto standard practice in software engineering is using version control systems such as Github to create an online copy of the source code, which can also include parameters and documentation. Similar version control systems exist for datasets. Public repositories can store experimental data on demand for significant amounts of time, enabling a long-term reproducibility of the experiment. Existing software engineering tools can be easily used to address many of the DOME recommendations. Although having additional, more topic-specific recommendations in the future will undoubtedly be useful, in this work we aim to provide a first version that could be of general interest. Adapting the DOME recommendations to address the unique aspects of specific topics and domains, would be a task of those particular (sub-)communities. For example, having guidelines for data independence is tricky because each biological domain has its own set of guidelines for this. Nonetheless, we believe it is relevant to at least have a recommendation that authors describe how they achieved data split independence. Discussions on the correct independence strategies are needed for all of biology. Given constructive consultation processes with ML communities, relying on our own experience, it is our belief that our manuscript can be useful as a first iteration of the recommendations for supervised ML in biology. This will have the additional benefit of kickstarting the community discussion with a coherent but rough set of goals, thus facilitating the overall engagement and involvement of key stakeholders. For instance, topics to be addressed by (sub-)communities is how to adapt DOME to entire pipelines, unsupervised, semi-supervised, reinforcement, and other types of ML. E.g. in unsupervised learning, the evaluation metrics shown in Figure 2 would not",
    "metadata": {
      "paper_id": "2006.16189v4",
      "chunk_index": 7,
      "total_chunks": 16
    },
    "distance": 1.1086560487747192,
    "rank": 2
  },
```

```
    {
      "chunk": "selecting the most informative emails in real time. Another application of online active learning in the field of IT has been recently presented by Zhang et al (2020a). They analyzed the scenario of network protocol identification and proposed a method (presented in Section 3.2) to select the most representative instances on the fly and adapt the model to dynamic data distributions. Stream of emails Stream of emails Receive an email Query label? Filter Classify email Update filter Yes No Fig. 13 Low-cost active spam filtering (Sculley, 2007). Computer vision is another interesting area where online active learning can be applied. Deep learning models require a large amount of annotated data, making manual annotation of thousands of images one of the most challenging aspects of model development. However, it is important to note that the most effective deep active learning methods proposed so far are not easily adaptable to a stream-based setting. Many of these methods involve clustering or measuring pairwise similarity among image embeddings (Sener and Savarese, 2017; Agarwal et al, 2020; Ash et al, 2019; Citovsky et al, 2021; Prabhu et al, 2020), which cannot be easily done in a single- pass manner. As a result, most online applications of active learning in computer vision rely on the use of traditional models with uncertainty-based sampling. Narr et al (2016) analyze the stream-based active learning problem for the classification of 3D objects. They used a mondrian forest classifier (Lakshminarayanan et al, 2014), which is an efficient alternative of random forest for the online learning scenario, and selected images with high classification uncertainty to be labeled. Ro˘zanec et al (2022) used online active learning to reduce the data labeling effort while performing vision-based process monitoring. Initially, features are extracted from the images using a pre-trained ResNet-18 model (He et al, 2015) and then, using the mutual information criterion (Kraskov et al, 2004), only √n features (Hua et al, 2005) are retained to fit an online classifier, where n is the total number of observations in the training set. The authors combine a simple active learning strategy based on model uncertainty with five streaming classification algorithms, including Hoeffding tree (Hulten et al, 2001), Hoeffding adaptive tree (Bifet and Gavald`a, 2009), stochastic gradient tree (Gouk et al, 2019), streaming logistic regression, and streaming k-nearest neighbors. Recently, Saran et al (2023) proposed a novel approach to streaming active learning with deep neural networks. Given a neural network with f with parameters θ, last- layer parameters θL, and the cross-entropy function ℓ, they compute the gradient representation of the data point xt, which is given by g(xt) = ∂ ∂θL ℓ(f(xt; θ), byt) (46) where byt = argmax f(xt; θ). Then, the data points to be included in the batch for training the model are chosen by using a probability pt proportional to the contribution of the current example to the covariance matrix of the examples collected so far, as in pt ∝det \u0010 bΣt + g(xt)g(xt)T\u0011 (47) where bΣt is the covariance matrix of the data points that have been selected to be included int he",
      "metadata": {
        "paper_id": "2302.08893v4",
        "chunk_index": 45,
        "total_chunks": 64
      },
      "distance": 1.118329644203186,
      "rank": 3
    }
  ]
}
```