Name: Karthik Hubli
UID:01689129

Literature Review-2(COMP-5460 Spring 2018)

The paper chosen for the review is 'Visualizing the Hiden Activity of Artificial Neural Networks' [Paulo E. Rauber , Samuel G. Fadel , Alexandre X. Falcão , Alexandru C. Telea] published in IEEE Transactions on Visualization and Computer Graphics (Volume: 23, Issue: 1, Jan. 2017) on 10 August 2016. The primary reference of this paper is 'Visualizing distortions and recovering topology in continuous projection techniques' [M. Aupetit].The reference is critical to the paper as it provides a stable and viable solution to the problem of visualization of large and continuous data which is part of the main paper, i.e. the large amount of states in a hiden layer of neural networks.

In machine learning, advances in computing power and techniques for building and training (deep) artificial neural networks (ANNs) have allowed these models to achieve state-of-the-art results in many applications related to pattern recognition. In data visualization, dimensionality reduction has been successfully used to compute projections: representations of high-dimensional data in lower-dimensional (usually 2D) spaces that try to preserve the data structure. The projection-based visualization approach that we propose for T1 is (sparsely) found in the machine learning literature. Specifically, using three widely studied benchmark image classification datasets, we show how our visualization approach is able to confirm facts that are already known about ANNs, and reveal previously unseen relationships between learned representations. A convolutional layer receives as input a image with c color channels, and connects each of its neurons to a small window (all channels included) of the input. The weights and biases of an ANN, which we collectively denote by a vector , are adapted to minimize a cost function C that penalizes prediction errors on the dataset . For example, a softmax output layer is typically combined with a negative log-likelihood cost function View Source Right-click on figure for MathML and additional features.

Machine learning experts have developed many strategies to design and improve ANNs, since the success of these models is highly impacted by the choice of preprocessing steps and several (interacting) hyperparameters. Visual analytics and information visualization systems have been developed to inspect ANNs, since visual feedback is considered highly valuable by practitioners.

In contrast to these works, our work is the first (to our knowledge) to present a detailed analysis of the insights on classification systems obtainable by projections of hidden layer activations. However, in contrast to such previous work, which explores relationships between input features (dimensions) to a pattern classification technique, we visualize relationships between features (neuron activations) learned by such a technique. Our visualization approach is based on hidden layer activations extracted from a network trained for a given dataset, and can be divided into two parts: creating projections from these activations (T1), and depicting the relationships between the neurons that originate these activations (T2). This section details the protocol followed by the experiments presented in Secs, 5 and 6, which simultaneously detail and evaluate our visualization approach.Datasets include three well-known image classification benchmarks: MNIST [22], SVHN [31] and CIFAR-I0 [20]. MNIST has 50K training images, 10K validation images, and 10K test images (grayscale images of handwritten digits). SVHN has 63.2K training images, 10K validation images, and 26K test images (color images of house number digits). CIFAR-10 has 30K training images, 10K validation images and 10K test images (color photographs in ten object classes). Although the images in SVHN and CIFAR-10 are quite small, which allows fast experimentation, these are not toy datasets, and are widely used to evaluate state-of-the-art ANNs.

Training is performed by momentum-based mini-batch stochastic gradient descent [5]. For MLPs, the batch size is 16, learning rate is 0.01, momentum coefficient is 0.9, and learning decay is 10−9. For CNNs, the batch size is 32, learning rate is 0.01, momentum coefficient is 0.9, and learning decay is 10−6. Initial weights for a neuron in layer l are sampled from a uniform distribution on , where , and biases start at 0. We manually chose these hyperparameters, together with the aforementioned architectures' based on cross-validation using the pre-defined validation sets. After the hyperparameters were chosen, we trained the models again using all data except the pre-defined test sets.Table summarizes the test set accuracy (AC, fraction of correctly classified observations) of our networks, and compares it to state-of-the-art networks, some of which also use preprocessing and data augmentation. Clearly, our networks achieve good accuracy on benchmark datasets. As such, they should be seen as realistic from an application perspective.

Dataset \ Model	MLP	CNN	State-of-the-art
MNIST	98.52%	99.62%	99.79% [47]
SVHN	77.38%	93.76%	98.08% [23]
CIFAR-10	52.91%	79.19%	91.78% [23]

Activations for a given layer, the subject of our analysis, are extracted for a random subset of 2000 observations from the test sets, strictly to facilitate visual presentation. For a given k (in our work,), the NH for a point is the ratio of its k-nearest neighbors that belong to the same class as . The NH for a whole projection is the average NH over its points. This is confirmed by the clear visual separation between classes in the (raw data) projection of a subset of 2000 test observations (784-dimensional vectors), shown in Fig. It is reasonable to hypothesize that a projection of the hidden layer activations of this MLP would have a significantly poorer visual separation between classes than the one shown in the projection. Hence, the learning process definitely arrived at an alternative representation of the data that captures class structure, which is reflected by the projection. 3b shows how an image of the digit 3 is (understandably) mistaken for the digit 5, and placed near the visual cluster corresponding to digit 5. This example shows that, despite the fact that projections may sometimes not fully preserve the data structure, as we discuss in Sec. The SVHN dataset is much more challenging for classification than MNIST. This is reflected, before training, in the visual separation between classes in the projection of the last hidden layer activations of an MLP, which is considerably poorer for SVHN (Fig. Consider the projection of the activations of the first MLP hidden layer after training (Fig. For instance, confusion matrices could be used to diagnose the confusion between classes for a learning algorithm trained on the hidden layer activations.

Upon further inspection (brushing points), we found that one of the same-colored visual clusters corresponds to dark digits on light backgrounds, and the other to light digits on dark backgrounds (see examples in Fig. To evaluate this, we preprocessed the images in SVHN in a simple manner: we apply the Sobel operator, after a small Gaussian blur, to approximate the gradient magnitude of the grayscale counterpart to each image. This yields grayscale images that are bright on the edges between background and foreground, and avoids the task

of detecting if a digit is light or dark, which is not trivial given the high variability of the images. For training data, it is natural to expect that the visual separation between classes will be even better than in test data. Firstly, a badly separated training set projection may indicate a poorly trained network, which has low chances of performing well on test data. A very well separated training set projection and a poorly separated test set projection may indicate poor generalization (caused, for example, by overfitting). Comparing projections from different architectures is also insightful: in our case, we see that the CNN performs considerably better on the training set than the MLP, which matches the perceived visual separation and NH for Figs. Although the CNN assigns the correct class to these points (digit 7), the representation of the last hidden layer places them near orange points. Since the class in cyan corresponds to truck images, and the outlier observation (automobile) looks very similar to members of that class, it is not surprising that the corresponding point becomes a visual outlier given the learned representation. Evolution of learned representations The previous sections presented projections of learned representations (activations) for combinations of datasets, training stages (epochs), and layers.

Secondly, considering the activation projections, although our particular choice of technique (Barnes-Hut t-SNE) is computationally scalable, it still requires approx. It is extremely important to address a specific threat to the validity of our approach: the fact that dimensionality reduction techniques provide few quality guarantees, and may introduce misleading visual artifacts. If a projection (or some of its parts) has poor quality, it should be discarded from further use. Finally, we note that the feedback given by (activation) projections for classification problems is, in a sense, asymmetric: clear visual separation between classes surely implies an easy classification task, whereas unclear separation does not necessarily imply a difficult task.

In this paper, we have shown how dimensionality reduction can be used to visualize the relationships between learned representations and between neurons in artificial neural networks. Concerning the first task, our visualizations support the identification of confusion zones, outliers, and clusters in the internal representations computed by such networks. They include visualizing representations learned by recurrent networks, which currently achieve state-of-the-art results in many sequence-related tasks