The LESCI Layer

Timo Denk* mail@timodenk.com

Florian Pfisterer* florian.pfisterer@me.com

November 2018

1 Notation

Let $f : \mathbb{X} \to \mathbb{Y}$ be a classifier, where $\mathbb{X}$ is the set of possible inputs and $\mathbb{Y}$ the set of labels. Each input is associated to exactly one label.

Using a neural network with N layers to represent f, we can examine different intermediate layers. We refer to the outputs of these layers as activations or representations. The ith layer is denoted as $f_i : \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ where $i \in \{1, \dots, N\}$. f_1 is the input layer and f_N is the output *softmax* layer that outputs a probablitity distribution over the labels in $\mathbb{Y}$.

After training a classifier on a dataset of tuples $\mathbb{X} \times \mathbb{Y}$, we insert a new layer l in between two existing layers f_j and f_{j+1}. We suggest multiple kinds of layers, introduced in the following.

2 Vector Quantization

We define a vector quantization (VQ) layer function $l_{\text{VQ}} : \mathbb{R}^m \to \mathbb{R}^m$ that is associated to an embedding space $\boldsymbol{E} \in \mathbb{R}^{n \times m}$. All inputs to this layer are discretized into one of the rows vectors $\boldsymbol{E}_{i,:}$. This idea has been used in an auto encoder by van den Oord et al. (2017), though we use it in another context here.

The VQ layer takes an input vector $\boldsymbol{x} \in \mathbb{R}^m$ and compares it to all vectors in $\boldsymbol{E}$ unsing a distance function $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$. It maps the input to the embedding space vector that is found to be most similar. The layer function is defined as

$$\boldsymbol{l}_{\text{VQ}}(\boldsymbol{x}) = \boldsymbol{E}_{i^*,:}, \tag{1}$$

*Equal contribution.

where i^* is given by

$$i^* = \arg\min_i d\left(\boldsymbol{E}_{i,:}, \boldsymbol{x}\right) . \tag{2}$$

Various function can be used for d, for instance the cosine similarity, where $d(\boldsymbol{x}, \boldsymbol{y}) = -\text{sim}(\boldsymbol{x}, \boldsymbol{y})$ and:

$$\text{sim}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{m} x_i \cdot y_i}{||x||_2 ||y||_2} \tag{3}$$

3 The LESCI Layer

"Large Embedding Space Constant Initialization" (LESCI) is an initialization technique for the embedding space of the VQ-layer. Here, we assume the VQ layer l_{VQ} is added in between two layers f_j and f_{j+1}. Its embedding matrix $\boldsymbol{E}$ is initialized with the outputs of f_j induced by feeding n samples from $\mathbb{X}$ through the network. We denote a VQ-layer that uses LESCI as its initialization method as $\boldsymbol{l}_{\text{LESCI}}$.

The intuition behind this initialization method is to store representation vectors associated with inputs for which the outputs are known, such that previously unseen samples will be projected to representations whose correct label is known. The following part of the network is then exclusively exposed to activation vectors that are known from the dataset that was used to initialize $\boldsymbol{E}$. All samples used to compute the representations with which $\boldsymbol{E}$ is initialized should be classified correctly by f.

Multiple LESCI layers can be applied to different parts of an activation vector, with shared or distinct embedding spaces.

3.1 Measuring Similarity after Dimensionality Reduction

Because the input $\boldsymbol{x}$ of a LESCI-layer is usually high-dimensional (e.g. in the image classification domain), measuring the distance using common distance functions d such as the L2-norm, L1-norm, or the cosine similarity defined above (3) may not result in activation vectors belonging to the same class actually having a high similarity as measured by d.

Dimensionality reduction techniques serve as a way to mitigate this problem by projecting both the input $\boldsymbol{x} \in \mathbb{R}^m$ as well as the embedding space $\boldsymbol{E} \in \mathbb{R}^{n \times m}$ down to a lower dimension $r << m$. Let $\hat{\boldsymbol{x}} \in \mathbb{R}^r$ be the lower-dimension input vector and $\hat{\boldsymbol{E}} \in \mathbb{R}^{n \times r}$ the lower-dimension embedding space. Then, the arg min in equation 2 is calculated over the lower-dimensional values $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{E}}$ as follows:

$$i^* = \arg\min_i d\left(\hat{\boldsymbol{E}}_{i,:}, \hat{\boldsymbol{x}}\right) . \tag{4}$$

Notice that d is evaluated here on r-dimensional vectors as opposed to m-dimensional vectors as before. Also note that the dimensionality-reduced vectors

are only used for choosing the closest embedding vector. The projection itself still comes from the original $\boldsymbol{E}$.

Principal Component Analysis (PCA) is a common technique for dimensionality reduction which we have employed.

3.2 Majority Vote

We extend $\boldsymbol{l}_{\text{LESCI}}$ with a majority vote that determines the classifier's output. Every vector $\boldsymbol{E}_{i,:}$ is associated with a particular label $l_i \in \mathbb{Y}$. For an input $\boldsymbol{x}$, we extract the top k nearest neighbors from $\boldsymbol{E}$, i.e. most similar vectors $\boldsymbol{E}_{i,:}$ as measured by d. The resulting vector of labels is denoted as $l_{\text{knn}} \in \mathbb{Y}^k$.

The most frequent label occuring in l_{knn} is chosen to be the classifier output if its number of occurrences o is exceeding a certain threshold, $\frac{o}{k} > t_{\text{projection}}$. $t_{\text{projection}}$ is a hyperparameter.

4 Intuition and Reasoning

We have developed the described methods to increase the robustness of an image classifier with respect to adversarial examples. Neural network classifiers are known to be vulnerable to such attacks, see Goodfellow et al. (2014). An adversarial attack is a slight modification of an input $\boldsymbol{x}$ yielding a new input $\tilde{\boldsymbol{x}}$ that is causing a misclassification.

The idea of LESCI-layers is to map slightly perturbed activation vectors back to values that are known to be classified correctly, thereby increasing the robustness of the network with respect to adversarial examples. The assumption is that slight changes in the input cause a slight change in the activation vectors, not significant enough to move the activation vector into an area where the k nearest neighbors are associated to different classes.

Other work, such as Liao et al. (2017), has analyzed the difference between the representations at some layer j for adversarial vs. clean images. Their findings show that this difference increases over the layers of the network. We conclude that placing $\boldsymbol{l}_{\text{LESCI}}$ early in the network results in adversarial inputs being mapped to the correct output label, making the network more robust.

However, the deeper a layer in a network, the more its representation contains information about the input's features and not about the input itself. Therefore, placing $\boldsymbol{l}_{\text{LESCI}}$ late in the network increases the expected accuracy of the projection.

Closer to the input, samples of the same class might differ more, while the perturbations are minor. Closer to the output, samples of the same class tend to be come more similar (until their probablitity distributions in the output layer

have the same arg min), while the perturbation caused by an adversarial input grows in magnitude. Thus, the location of the LESCI layer(s) in the network is a hyperparameter that balances accuracy (which increases when located late in the network) and robustness (which increases when located early in the network).

In general, an embedding space should be initialized with as many labeled and correctly classified samples as possible.

References

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Jun Zhu, and Xiaolin Hu. Defense against adversarial attacks using high-level representation guided denoiser. *CoRR*, abs/1712.02976, 2017. URL http://arxiv.org/abs/1712.02976.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *CoRR*, abs/1711.00937, 2017. URL http://arxiv.org/abs/1711.00937.