

Intuition Learning

Anush Sankaran¹, Mayank Vatsa², Richa Singh²

¹ IBM Research

² IIIT, Delhi, India

anussank@in.ibm.com, mayank@iiitd.ac.in, rsingh@iiitd.ac.in

Abstract

“By reading on the title and abstract, do you think this paper will be accepted in IJCAI 2018?” A common impromptu reply would be “I don’t *know* but I have an *intuition* that this paper might get accepted”. Intuition is often employed by humans to solve challenging problems without explicit efforts. Intuition is not trained but is learned from one’s own experience and observation. The aim of this research is to provide *intuition* to an algorithm, apart from what they are trained to *know* in a supervised manner. We present a novel intuition learning framework that learns to perform a task completely from unlabeled data. The proposed framework uses a continuous-state reinforcement learning mechanism to learn a feature representation and a data-label mapping function using unlabeled data. The mapping functions and feature representation are succinct and can be used to supplement any supervised or semi-supervised algorithm. The experiments on the CIFAR-10 database shows challenging cases where intuition learning improve the performance of a given classifier.

1 Introduction

Intuition refers to knowledge acquired without inference and/or the use of reason [Simpson *et al.*, 1989]. Philosophically, there are several definitions for intuition and the most popularly used one is “Thoughts that are reached with little apparent effort, and typically without conscious awareness” [Hogarth, 2001] and is considered as the opposite of a rational process. From a machine learning perspective, training a supervised classifier is a rational process where it is trained with labeled data allowing it to learn a decision boundary. Also, traditional unsupervised learning methods do not map the learnt patterns to their corresponding class labels. Semi-supervised approaches bridge this gap by leveraging unlabeled data to better perform supervised learning tasks. However, the final task (say, classification) is performed only by a supervised classifier using labeled data with some additional knowledge from unsupervised learning. The notion of intuition would mean that the system performs tasks using only unlabeled data without any supervised (rational)

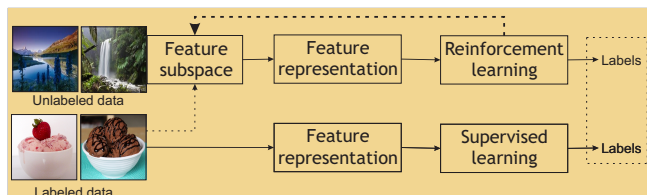


Figure 1: The proposed solution where an intuition learning algorithm can supplement a supervised or semi-supervised algorithm at decision level.

learning. In other words, intuition is a context dependent *guesswork* that can be incorrect certain times. In a typical *learning* pipeline, the concept of intuition can be used for a variety of purposes starting from training data selection upto and including decision making. Heuristics are the simplest form of intuition that bypass or is used in conjunction with rational decisions to obtain quick approximate results. For example, heuristics can be used in (1) choosing the new data points in an online active learning scenario [Chu *et al.*, 2011], (2) for feature representation [Coates *et al.*, 2011], (3) feature selection [Guyon and Elisseeff, 2003], or (4) choice of classifier and its parameters [Cavalin *et al.*, 2012].

Table 1 shows the comparison of existing popular machine learning paradigms. Supervised learning attempts to learn an input-output mapping function on a feature space using a set of labeled training data. Transfer learning aims to improve the target learning function using the knowledge in source (related) domain and source learning tasks [Pan and Yang, 2010]. Many types of knowledge transfer such as classification parameters [Lawrence and Platt, 2004], feature representations [Evgeniou and Pontil, 2007], and training instances [Huang *et al.*, 2006] have been tested to improve the performance of supervised learning tasks. Semi-supervised learning utilizes additional knowledge from unlabeled data, drawn from the same distribution and having the same task labels as the labeled data. Many of these research works has focussed on unsupervised feature learning i.e., to create a feature subspace using the unlabeled data, to which the labeled data can be projected to obtain a new feature representation [Chapelle *et al.*, 2006]. In 2007, Raina *et al.* [Raina *et al.*, 2007] proposed a framework termed as “Self-taught learning” to create the generic feature subspace using sparse auto-

Paradigm	Input data	Learnt function	Comments
Supervised [Bishop and Nasrabadi, 2006]	<data, label>	data-label mapping	
Unsupervised [Bishop and Nasrabadi, 2006]	<data>	data clusters	
Semi-supervised [Chapelle <i>et al.</i> , 2006]	<data, label>, unlabeled data	data-label mapping	unlabeled data follow the same distribution
Reinforcement [Kaelbling <i>et al.</i> , 1996]	reward function (or value)	state, action policy	need a teacher to provide reward
Active [Settles, 2010]	<data, label>	data-label mapping, new data selection	need human annotator (Oracle) or expert algorithm to provide labels for new data
Transfer [Pan and Yang, 2010]	<sourceData, sourceLabel>, <targetData, targetLabel>	targetData - targetLabel mapping	transfer can be data instances, classification parameters, or features
Imitation [Natarajan <i>et al.</i> , 2011]	sourceData, sourceData-sourceLabel mapping	targetData - targetLabel mapping	need a teacher to provide reward
Self taught [Raina <i>et al.</i> , 2007]	<data, label>, unlabeled data	data-label mapping	unlabeled data need not follow the same distribution and label as data
Deep learning [Bengio, 2009]	<data, label>, unlabeled data	data-label mapping	complex architecture to learn robust data representations
Intuition	data, unlabeled data, reward function (or value)	data-label mapping	unlabeled data need not follow the same distribution, need a reward function

Table 1: Comparison of existing popular machine learning paradigms along with the proposed intuition learning paradigm.

encoders irrespective of the task labels. Self-taught learning dismisses the same class label assumption of semi-supervised learning and forms a generic high-level feature subspace from the unlabeled data, where the labeled data can be projected.

As shown in Figure 1, we postulate a framework of supplementing *intuition decisions* at the decision level to a supervised or semi-supervised classifier. The decisions drawn by the reinforcement learning block in Figure 1 are called *intuition* because they are learnt only using the unlabeled data with an indirect reward from a teacher. Existing algorithms, broadly, require training labels for building a classifier or borrows the classifier parameters from an already trained classifier. Direct or indirect training is not always possible as obtaining data labels is very costly. To address this challenge, we propose a novel paradigm for unsupervised task performance mechanism learnt from cumulative experience. Intuition is modeled as a learning framework, which provides the ability to learn a task completely from unlabeled data. By using continuous state reinforcement learning as a classifier, the framework learns to perform the classification task without the need for explicit labeled data. Reinforcement learning helps in adapting a randomly initialized feature space to the specific task at hand, where a parallel supervised classifier is used as a teacher. As the proposed framework is able to learn a mapping function from the input data to the output class labels, without the requirement for explicit training, it functions similar to human intuition and we term this approach as *Intuition Learning*.

1.1 Research Contributions

This research proposes a intuition learning framework to enable algorithms learn a specific classification or regression task completely from unlabeled data. The major contributions of this research are as follows:

- A continuous state reinforcement learning based clas-

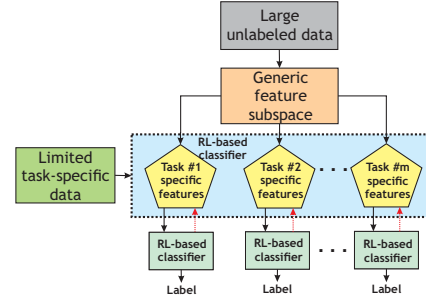


Figure 2: A block diagram outlining on how a feature space can be adapted using reinforcement learning algorithm with feedback from a supervised classifier trained on limited task-specific data.

sification framework is proposed to map input data to output class label, without the explicit use of training.

- A residual Q-learning based function approximation method for learning the feature representation of task specific data. A novel reward function which does not require class labels is designed to provide feedback to the reinforcement based classification system.
- A context dependent addition framework is proposed, where the result of the intuition framework can be supplemented based on the confidence of the trained supervised or semi-supervised mapping function.

2 An Intuition Learning Algorithm

The basic idea of the proposed intuition learning framework is presented in Figure 2. Given a large set of unlabelled data, different kinds of feature representations are extracted to describe the data, irrespective of the task in hand. To further leverage the knowledge interpretation from unlabeled data, a

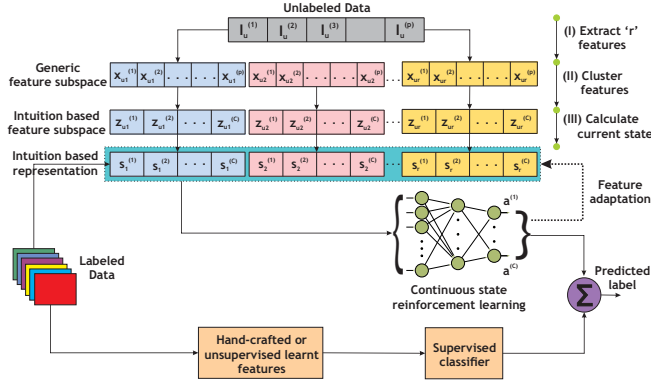


Figure 3: Overall scheme of the proposed intuition learning algorithm that aids a supervised classifier.

continuous state reinforcement learning mechanism is used to perform the given classification task. As reinforcement is a continuous learning process, using a reward based feedback mechanism, the classification task improves with time. The reinforcement learning, on one hand acts as a classifier, while on the other hand continuously adapts the feature representation with respect to the given task. Thus, given multiple tasks, the proposed intuition learning framework can adapt the generic feature space to be consistent with the corresponding task.

Let $\{(I_l^{(1)}, y^{(1)}), (I_l^{(2)}, y^{(2)}), \dots, (I_l^{(m)}, y^{(m)})\}$ be the set of m labeled training data drawn i.i.d. from a distribution D . The labeled data are represented as $\{(x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}), \dots, (x_l^{(m)}, y^{(m)})\}$, where $x_l^{(i)} \in R^n$ is the feature representation of the data $I_l^{(i)}$ and $y^{(i)} \in [1, 2, \dots, C]$ denotes the class label corresponding to $x_l^{(i)}$. Let the set of unlabeled data be $\{I_u^{(1)}, I_u^{(2)}, \dots, I_u^{(p)}\}$, where the subscript u represents that they are unlabeled data. Contrary to self-taught learning [Raina *et al.*, 2007], we do not assume that the labeled and unlabeled data should be drawn from the same distribution D or have the same class labels, however, they should be derived from the same modality. Given a set of labeled and large unlabeled data, the aim of intuition learning is to learn a hypothesis $h' : (X \rightarrow R) \in [1, 2, \dots, C]$ that predicts the labels for a given input representation of data drawn. However, the hypothesis h' is learnt without the direct use of labels $y^{(i)}$ and is used as a supplement for the hypothesis h learnt using $(x_l^{(1)}, y^{(1)})$ in a supervised (or semi-supervised) manner.

2.1 Adapting Feature Representation

From a large set of unlabeled data, many different kind of feature representations are extracted. Each representation may correspond to a different property of the data that we try to capture. For image data, the features could be color, texture, and shape while for text data, the features could be n-grams, bag-of-words, and word embeddings. The features can also be a set of different color features or set of hierarchical n-grams. If the large set of unlabeled data is seen as the world

(or the universal set), the features are the different observations made by the algorithm from the world. Similar to human intuition, the set of feature representations extracted are task independent, and later depending on the learning task a subset of these features could be dominantly used. This task independent feature space is similar to the human intuition learnt by observing the environment.

Figure 3 provides a detailed description of the proposed intuition learning framework. From the set of unlabeled data I_u , we extract r different kinds of feature representations, $\{X_{u1}, X_{u2}, \dots, X_{ur}\}$, where $X_{ui} = \{x_{ui}^{(1)}, x_{ui}^{(2)}, \dots, x_{ui}^{(p)}\}$, where $x_{ui}^{(j)} \in R^{n_i}$. For every feature representation $q \in [1, 2, \dots, r]$, we cluster the representation $[x_{uq}^{(1)}, x_{uq}^{(2)}, \dots, x_{uq}^{(p)}]$ into C clusters¹ using k-means clustering. The centroid of each cluster for the i^{th} feature representation is given as $[z_{u(i)}^1, z_{u(i)}^2, \dots, z_{u(i)}^C]$. This feature collection of $[z_{u(q)}^1, z_{u(q)}^2, \dots, z_{u(q)}^C]$, for $q = [1, 2, \dots, r]$ is called as *Intuition based Feature Subspace (IFS)*, as it clusters the entire set of unlabeled data into groups, based on every observation (feature).

2.2 Classification using Reinforcement Learning

For a given set of m labeled training data, $\{I_l^{(1)}, I_l^{(2)}, \dots, I_l^{(m)}\}$, the set of r features (as used for the unlabeled data) are extracted as $[x_{lq}^{(1)}, x_{lq}^{(2)}, \dots, x_{lq}^{(m)}]$, where $q = [1, 2, \dots, r]$. The extracted features are then projected onto the Intuition based Feature Subspace (IFS) by calculating the distance of features from the corresponding cluster centroids shown as,

$$s_q^{(i)} = \|x_{lq}^{(i)} - z_{uq}^{(j)}\|_2 \quad (1)$$

for $j = [1, 2, \dots, C]$, $q = [1, 2, \dots, r]$, and $i = [1, 2, \dots, m]$. The representation of the data i is given by concatenating the distances corresponding to all the features,

$$s^{(i)} = [s_1^{(i)}, s_2^{(i)}, \dots, s_r^{(i)}] \quad (2)$$

The obtained representation is succinct with a fixed length dimension of $rC \times 1$, where r is the number of different feature types extracted and C is the number of clusters. In essence, every value represents the distance from a cluster centroid. Also, in a typical semi-supervised (or self-taught) learning scheme, the mapping between intuition based representation and the output class labels, $\{(s^{(1)}, y^{(1)}), (s^{(2)}, y^{(2)}), \dots, (s^{(m)}, y^{(m)})\}$ is learnt in a supervised manner. However, in the proposed intuition learning, we attempt to learn the data-label mapping without using the class labels, using reinforcement learning. The aim of reinforcement learning is to learn an action policy $\pi : s \rightarrow a$, where $s \in S$ is the current state of the system and a is the action performed in that state. As the setup involves a continuous state environment, the optimal action policy is learnt using a model free, off-policy Temporal Difference (TD) algorithm called *Q-learning*, where $Q(s, a)$ -value denotes the

¹The best adaption results are obtained when we fix C to be the number of classes we have in the learning task

effectiveness of a state-action pair. The $TD(0)$ Q-learning algorithm is given by,

$$Q(s_t, a) = Q(s_t, a) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a) \right] \quad (3)$$

where, $r_t \in R^n$ is the immediate reward obtained for performing action a in state s_t , $\gamma \in [0, 1]$ is the factor with which the future rewards are discounted and $\alpha \in [0, 1]$ is the learning rate. In our problem, reinforcement learning is formulated as a classification problem, where IFS is the current state s and action a is the output label to be predicted, the policy π learns the data-label relation for the given data. Due to the large, probabilistic, and continuous definition of the space s , the Q-values are approximated using a universal function approximation ie., a neural network [Sutton and Barto, 1998].

$$Q(s, a) = \psi(s, a, \theta) = \sum_i \phi_i(s, a) \cdot \theta_i = \phi^T(s, a) \cdot \theta \quad (4)$$

where, ϕ is the approximation function. Using residual Q-learning algorithm [Paletta and Pinz, 2000], the free parameters θ are updated as follows,

$$\theta_{t+1} = \theta_t + \alpha \cdot \psi \cdot \Delta \psi \quad (5)$$

$$\begin{aligned} \theta_{t+1} = \theta_t + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a) \right] \\ \times \left[\beta \gamma \frac{\partial}{\partial \theta} \max_{a'} Q(s_{t+1}, a') - \frac{\partial}{\partial \theta} Q(s_t, a) \right] \end{aligned} \quad (6)$$

where β is a weighting factor called the Bellman residual. Baird [Baird, 1995] guaranteed the convergence of the above approximate Q-learning function, the details of which are skipped for the sake of brevity. ϵ -exploration strategy is adopted, where, in every state a random action is preferred with a probability of ϵ . As observed in [Lagoudakis and Parr, 2003], “the crucial factor for a successful approximate algorithm is the choice of the parametric approximation architecture and the choice of the projection (parameter adjustment) method(s)”. The choice of reward function employed is highly important and directly implies the effectiveness of adaption, which is explained in the next section.

2.3 Design of Reward Function

The Intuition based Feature Subspace (IFS) is defined by the cluster centroid points obtained using unlabeled data for every feature q as $[z_{u_q}^{(1)}, z_{u_q}^{(2)}, \dots, z_{u_q}^{(C)}]$, where $q = [1, 2, \dots, r]$. This space provides an organized definition of how the entire set of unlabeled data is observed and inferred. From the various features of the labeled training data $[(x_{l_q}^{(1)}, y^{(1)}), (x_{l_q}^{(2)}, y^{(2)}), \dots, (x_{l_q}^{(m)}, y^{(m)})]$, where $q \in [1, 2, \dots, r]$, the centroid points for every feature and every class are calculated as, $[z_{l_q}^{(1)}, z_{l_q}^{(2)}, \dots, z_{l_q}^{(C)}]$, where $q = [1, 2, \dots, r]$. This space, called the *Labeled data Feature Subspace (LFS)*, formed by these centroid points provide us the inference of the particular learning task to be performed. It is to be noted that:

- Apart from unlabeled data, every labeled training data (and even testing data) gets incrementally added to the IFS , as the observed data affects the overall understanding of features.
- The aim of this incremental learning, is to shape the IFS as close as possible to LF while learning the feature-label mapping function using reinforcement learning.

The incremental update of the IFS happens for the i^{th} training example belonging to j^{th} class, as shown in the following equation,

$$z_{u_q}^{(j)} = z_{u_q}^{(j)} + \left(\frac{x_{l_q}^{(i)} - z_{u_q}^{(j)}}{n_q^j} \right) \quad (7)$$

for $q = [1, 2, \dots, r]$, where n_q^j is the number of data points in the j^{th} cluster for q^{th} feature. Further, to make effective learning from this incremental update, the reward function is defined as a function of the distance between the current IFS and LFS , as follows:

$$r_t = \left(\|z_{u_q, t}^{(j)} - z_{l_q}^{(j)}\|_2 \right)^{-1} \quad (8)$$

for $q = [1, 2, \dots, r]$, $j = [1, 2, \dots, C]$ at a given time t .

2.4 Context Dependent Addition Mechanism

Intuition learning framework acts as a supplement to (and not complementing) supervised learning. The need for intuition arises only when the confidence of supervised learner falls below a particular threshold. Therefore, a context dependent mechanism is designed to leverage supervised learning using intuition only when required. For given labeled training data $\{I_l^{(1)}, I_l^{(2)}, \dots, I_l^{(m)}\}$, some hand-written or unsupervised features are extracted, $\{(x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}), \dots, (x_l^{(m)}, y^{(m)})\}$ and a supervised model is learnt, $H_s : (x_l^{(i)} \rightarrow \hat{y}_s)$. Based on the supervised learning algorithm, the classification confidence is computed for the i^{th} data point and is given as $conf_s^{(i)} = [cs_1^{(i)}, cs_2^{(i)}, \dots, cs_C^{(i)}]$. The mechanism to calculate the classification confidence depends on the supervised learning model used. Similarly, the intuition learning can be represented as $H_{int} : (s^{(i)} \rightarrow \hat{y}_{int})$ and the classification confidence is the output of the last layer of the value function approximation neural architecture, given as $conf_{int}^{(i)} = [cint_1^{(i)}, cint_2^{(i)}, \dots, cint_C^{(i)}]$. A label switching mechanism is performed to give the final predicted label, \hat{y} , as follows,

$$\hat{y} = \begin{cases} \hat{y}_s, & \Delta > th \\ \hat{y}_{new}, & \text{otherwise} \end{cases} \quad (9)$$

where th is the threshold for using intuition and the condition for context Δ is calculated as follows,

$$\Delta = \max_j (cs_j^{(i)}) - \max_{l \neq j} (cs_l^{(i)}) \quad (10)$$

Type	Feature	Dimension
Color	Color Harris [Van De Weijer <i>et al.</i> , 2006]	10×2
Color	Color Autocorrelograms [Huang <i>et al.</i> , 1997]	64×1
Local Texture	Local Binary Pattern (LBP) [Ojala <i>et al.</i> , 2002]	59×4
Global Texture	GIST [Oliva and Torralba, 2001]	512×1
Saliency	Region covariances [Erdem and Erdem, 2013]	32×32
Shape	Multilayer autoencoder [Vincent <i>et al.</i> , 2008]	10×1

Table 2: Details of different features extracted from the image data.

In such cases where intuition is used to boost the confidence of supervised classifier the new label is computed as follows:

$$cnew_k^{(i)} = \lambda \cdot cs_k^{(i)} + (1 - \lambda) \cdot cint_k^{(i)} \quad (11)$$

$$\hat{y}_{new} = \arg \max_j (cnew_j^{(i)}) \quad (12)$$

where λ is the trade-off parameter between intuition and supervised learning. Thus, in simple words, we add the feeling of intuition to an algorithm.

3 Experimental Analysis

3.1 Dataset

The proposed intuition learning algorithm is applied for 10-class classification problem using the CIFAR-10 database [Krizhevsky and Hinton, 2009]. The database contains 60,000 color images labeled, each of size 32×32 pertaining to 10 classes (i.e. 6,000 images per class). There are 50,000 training images and 10,000 test images. The data set contains small size images, leading to limited and noisy information content and it provides the most relevant case study to demonstrate the effectiveness of the proposed paradigm. The STL-10 database [Coates *et al.*, 2011] is used as the unlabeled image data set having one million colored images of size 96×96 . As shown in Table 2 six different feature representation are extracted from the images. These features comprehensively comprise the various types of features that could be extracted from image data. For all the experiments, five times random cross-validation is performed and the best model accuracy is reported for all the experiments.

3.2 Interpreting Intuition based Feature Subspace

The primary aim of the approach is to construct the feature subspace completely from unlabeled data and to adapt it to a specific learning task. Figure 4 shows the clusters of entire unlabeled data corresponding to every feature extracted. The concatenation of the feature spaces put together in Figure 4(a) represents the *IFS*. Figure 4(b) shows the adapted task-specific feature subspace after performing 300 epochs of learning with the given labeled data. Figure 4(c) shows the amount of update in the cluster after adding an image, by calculating the dissimilarity between the cluster centroid, before and after the addition of the image. Cluster dissimilarity is calculated for the r -th feature representation as follows:

$$C_{dis} = \sum_{j=1}^C \frac{1}{D_j} \cdot ||z_{u_r}^{(j)}|_{(t+1)} - z_{u_r}^{(j)}|_{(t)}||_2 \quad (13)$$

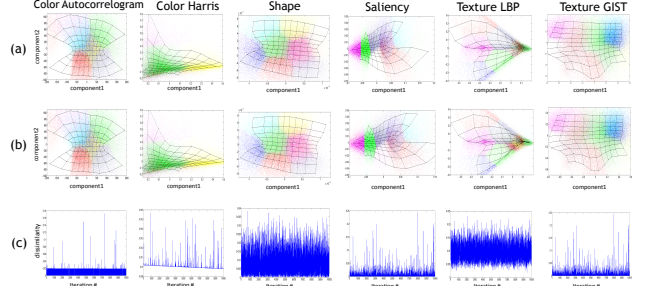


Figure 4: Image showing the data clusters for each of the extracted feature and grid depicts the cluster density at local regions. (a) shows the *IFS* of all the unlabeled data, (b) shows the adapted task specific feature space after 300 epochs of learning, and (c) shows the amount of change happening in the cluster after the addition of an image. Best viewed in color.

Training size	Supervised	Semi supervised	Supervised intuition
10000	38.90	29.64	36.19
5000	37.13	24.34	34.78
3000	14.12	17.07	25.65
1000	10.34	16.54	19.61

Table 3: The performance accuracy (%) of supervised intuition learning is compared with supervised (neural network) and semi-supervised (self-taught) learning methods. The significance of intuition, is studied by varying the amount of available training data. 5 times random cross validation is performed and the best model's performance is reported.

where D_j is the density of the j^{th} cluster. It can be visually observed from the plot that, shape, gist, and LBP feature spaces are updated (learns) after the addition of each image, indicating that these features contribute more towards the classification task. However, both color Harris and autocorrelogram features are not much updated by the training data.

3.3 Performance Analysis

It is to be noted that intuition learning framework is used to supplement any supervised or semi-supervised learning mechanism. In this paper, we show the results in the following scenario:

1. Using two supervised learning algorithms (back-propagation neural network and multi-class SVM) with Uniform Circular Local Binary Pattern (UCLBP) [Ojala *et al.*, 2002] as features. Labeled data, $[(x_{l_q}^{(1)}, y^{(1)}), (x_{l_q}^{(2)}, y^{(2)}), \dots, (x_{l_q}^{(m)}, y^{(m)})]$, from

Training size	Intuition	Intuition supervised	Intuition semi-supervised
10000	12.11	36.19	29.21
5000	10.00	34.78	23.85
3000	10.00	25.65	22.49
1000	08.99	19.61	20.53

Table 4: The influence of supplementing intuition to supervised and semi-supervised algorithm is shown by improvement in the performance accuracy (%).

Training size	Supervised	Intuition supervised
10000	44.57	41.83
5000	43.21	40.77
3000	13.56	17.48
1000	06.19	09.78

Table 5: The performance accuracy (%) of supervised and supervised intuition framework using SVM classifier is studied.

CIFAR-10 is used to train the supervised algorithms.

2. Using a semi-supervised learning algorithm, with neural network as classifier and UCLBP features trained on CIFAR-10 dataset. The semi-supervised algorithm used for comparison is one approach for self-taught learning [Raina *et al.*, 2007], with unlabeled data from STL-10 dataset, $\{(s^{(1)}, y^{(1)}), (s^{(2)}, y^{(2)}), \dots, (s^{(m)}, y^{(m)})\}$.
3. Using a intuition learning framework only, having the intuition based task specific feature representation combined with a continuous state reinforcement learning (Q-learning) in Equation 4 for classification.
4. Using a supervised intuition framework, where the output of the supervised learning algorithm and the intuition learning framework is combined using the context dependent addition mechanism in Equation 12.
5. Using a semi-supervised intuition framework, where the output of the semi-supervised learning algorithm and the intuition learning framework is combined using the context dependent addition mechanism.

The optimized values of various parameters used in our framework are as follows: $\alpha = 0.99, \gamma = 0.95, \beta = 0.2, th = 0.9, \lambda = 0.5$, and $\epsilon = 0.05$. Preprocessing of features is done using z-score normalization. All the experiments are performed on a Intel Xeon E5 – 2640 0, 2.50GHz, 64GB RAM server.

As already discussed, intuition has a better significance in challenging problems with limited training data. Table 3, Table 4, and Table 5 show the performance of the proposed intuition learning in comparison with other learning methods, by varying the training size as parameter². It can be observed that with enough training data, supervised algorithms (both neural network and SVM) yield the best classification performance. However, with decrease in the size of training data, the performance of all the three algorithms, supervised, semi-supervised, and intuition learning reduces. The results show that in such a scenario, incorporating intuition with supervised or semi-supervised algorithm yields improved results.

²For a given training size, the same subset of images is used across all the classifiers to avoid any training bias.



Figure 5: Examples of (a) success and (b) failure cases of the proposed intuition learning. AL = actual ground truth label, SL = label predicted by the supervised neural network learner, and IL = label predicted when intuition is combined with supervised neural network learner.

This supports our hypothesis that adding intuition would improve the performance from under challenging circumstances such as limited training data. Similarly from a human’s perspective, under the presence of all data and information, one may take correct decisions. However, when the background training data information is limited, intuition learning helps. Further, some key analysis are summarized below:

1. To study the effectiveness of residual learning in Equation 6, training error over successive epochs is plotted, for a training size of 10000. It can be observed that the training error gradually decreases and remains constant after 300 epochs, indicating that maximum training capacity has been achieved, with minimum training error.
2. The computation time required for intuition learning depends on the complexity of r features that are extracted. However, for one sample, under the assumption that the feature extraction happens off-line, the overall intuition decision and feature space can be generated in 0.082s while the supervised decision can be taken in ~ 4 s on an average. This shows that intuition is much faster requiring little effort than supervised decision making.
3. In Figure 5, some success and failure example cases are shown where (a) intuition helps in correctly classifying a data but supervised learning fails and (b) data was incorrectly classified because of intuition. As previously discussed, intuition can go wrong sometimes. Upon analyzing the first horse example in failure case (Figure 5(b)), it is observed that horses are clustered more towards brown color in the autocorrelogram color feature space. However, as the horse shown in the images is white in color, it gets clustered along with cat and misclassified by intuition learning.

4 Conclusion

Inspired from human capabilities of instinct reasoning, this research presents a intuition learning framework that supplements a classifier for improved performance, especially with limited training data. Intuition is modeled as a continuous state reinforcement learning, that adapts to a particular task using large amount of unlabeled data and limited task specific data. The performance of intuition is shown in a 10 class image classification problem, in comparison with supervised, semi-supervised, and reinforcement learning. The results indicate that the application of intuition improves the performance of the classifier with limited training data.

References

- [Baird, 1995] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *ICML*, pages 30–37, 1995.
- [Bengio, 2009] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.
- [Bishop and Nasrabadi, 2006] Christopher M Bishop and Nasser M Nasrabadi. *PRML*, volume 1. 2006.
- [Cavalin *et al.*, 2012] Paulo R Cavalin, Robert Sabourin, and Ching Y Suen. LoGID: An adaptive framework combining local and global incremental learning for dynamic selection of ensembles of HMMs. *PR*, 45(9):3544–3556, 2012.
- [Chapelle *et al.*, 2006] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006.
- [Chu *et al.*, 2011] Wei Chu, Martin Zinkevich, Lihong Li, Achint Thomas, and Belle Tseng. Unbiased online active learning in data streams. In *ACM SIGKDD*, pages 195–203, 2011.
- [Coates *et al.*, 2011] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *ICAIS*, pages 215–223, 2011.
- [Erdem and Erdem, 2013] Erkut Erdem and Aykut Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of vision*, 13(4), 2013.
- [Evgeniou and Pontil, 2007] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. In *NIPS*, volume 19, page 41, 2007.
- [Guyon and Elisseeff, 2003] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, 2003.
- [Hogarth, 2001] Robin M Hogarth. *Educating intuition*. University of Chicago Press, 2001.
- [Huang *et al.*, 1997] Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *CVPR*, pages 762–768, 1997.
- [Huang *et al.*, 2006] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *NIPS*, pages 601–608, 2006.
- [Kaelbling *et al.*, 1996] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *arXiv cs/9605103*, 1996.
- [Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master’s thesis, Department of Computer Science, University of Toronto*, 2009.
- [Lagoudakis and Parr, 2003] Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *JMLR*, 4:1107–1149, 2003.
- [Lawrence and Platt, 2004] Neil D Lawrence and John C Platt. Learning to learn with the informative vector machine. In *ICML*, pages 65–72, 2004.
- [Natarajan *et al.*, 2011] Sriraam Natarajan, Saket Joshi, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik. Imitation learning in relational domains: A functional-gradient boosting approach. In *IJCAI*, pages 1414–1420, 2011.
- [Ojala *et al.*, 2002] Timo Ojala, Matti Pietikainen, and Topi Maenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.
- [Oliva and Torralba, 2001] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.
- [Paletta and Pinz, 2000] Lucas Paletta and Axel Pinz. Active object recognition by view integration and reinforcement learning. *Robotics and Autonomous Systems*, 31(1):71–86, 2000.
- [Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):1345–1359, 2010.
- [Raina *et al.*, 2007] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, pages 759–766, 2007.
- [Settles, 2010] Burr Settles. Active learning literature survey. Technical report, Computer Sciences Technical Report 1648, University of WisconsinMadison, 2010.
- [Simpson *et al.*, 1989] John Andrew Simpson, Edmund SC Weiner, et al. *The Oxford english dictionary*, volume 2. Clarendon Press Oxford, 1989.
- [Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Cambridge University Press, 1998.
- [Van De Weijer *et al.*, 2006] Joost Van De Weijer, Theo Gevers, and Arnold WM Smeulders. Robust photometric invariant features from the color tensor. *IEEE TIP*, 15(1):118–127, 2006.
- [Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, 2008.