

SuperTML: Two-Dimensional Word Embedding and Transfer Learning Using ImageNet Pretrained CNN Models for the Classifications on Tabular Data

Baohua Sun¹, Lin Yang¹, Wenhan Zhang¹, Michael Lin¹, Patrick Dong¹, Charles Young¹ and Jason Dong¹

¹Gyr Falcon Technology Inc.

{baohua.sun}@gyrfalcontech.com

Abstract

Tabular data is the most commonly used form of data in industry. Gradient Boosting Trees, Support Vector Machine, Random Forest, and Logistic Regression are typically used for classification tasks on tabular data. DNN models using categorical embeddings are also applied in this task, but all attempts thus far have used one-dimensional embeddings. The recent work of Super Characters method using two-dimensional word embeddings achieved the state of art result in text classification tasks, showcasing the promise of this new approach. In this paper, we propose the SuperTML method, which borrows the idea of Super Characters method and two-dimensional embeddings to address the problem of classification on tabular data. For each input of tabular data, the features are first projected into two-dimensional embeddings like an image, and then this image is fed into fine-tuned two-dimensional CNN models for classification. Experimental results have shown that the proposed SuperTML method had achieved state-of-the-art results on both large and small datasets.

1 Introduction

In data science, data is categorized into structured data and unstructured data. Structured data is also known as tabular data, and the terms will be used interchangeably. DNN models are widely applied for usage on unstructured data such as image, speech, and text. Anthony Goldbloom, the founder and CEO of Kaggle observed that winning techniques have been divided by whether the data was structured or unstructured [Vorhies, 2016]. According to Anthony, “When the data is unstructured, it’s definitely CNNs and RNNs that are carrying the day” [Vorhies, 2016]. On the other side of the spectrum, machine learning models such as Support Vector Machines (SVM), Gradient Boosting Trees (GBT), Random Forest, and Logistic Regressions, have been used to process structured data. Regarding structured data competitions, Anthony says that XGboost is winning practically every competition in the structured data category [Fogg, 2016]. XGBoost [Chen and Guestrin, 2016] is one popular package implementing the Gradient Boosting method. Other implemen-

tations and improvements include lightgbm [Ke *et al.*, 2017], and catboost [Prokhorenkova *et al.*, 2018]. According to a recent survey of 14,000 data scientists by Kaggle (2017), a subdivision of structured data known as relational data is reported as the most popular type of data in industry, with at least 65% working daily with relational data. In this paper, we address the classification problem for structured data, which we call TML (Traditional Machine Learning) for convenience in this paper.

Recent research introduces RNNs and one-dimensional CNNs for the TML problems [Lam *et al.*, 2018; Thomas, 2018], and also categorical embedding for tabular data with categorical features [Guo and Berkahn, 2016; Chen *et al.*, 2016]. The categorical embeddings used in these DNN models are one-dimensional. One-dimensional embedding is widely used in NLP tasks, which projects each token into a vector containing numerical values, for example, a one-dimensional embedding word vector with the shape of 300x1. However, in recent NLP research, two-dimensional embedding of the Super Characters method [Sun *et al.*, 2018a] achieves state-of-the-art results on large dataset benchmarks. The Super Characters method is a two-step method for the text classification problem. In the first step, the characters of the input text are drawn onto a blank image, so that an image of the text is generated with each of its characters embedded by the pixel values in the two-dimensional space, i.e. a matrix. The resulting image is called the Super Characters image. In the second step, the generated Super Characters image is fed into a two-dimensional CNN models for classification. The two-dimensional CNN models are trained for the text classification task through the method of Transfer Learning, which finetunes the pretrained models on large image dataset, e.g. ImageNet [Russakovsky *et al.*, 2015], with the labeled Super Characters images for the text classification task.

Many successful two-dimensional CNN models have been introduced in the ImageNet competition, such as ResNet [He *et al.*, 2016], SENet [Hu *et al.*, 2018], PolyNet [Zhang *et al.*, 2017], and NASNet [Zoph *et al.*, 2018]. Current state of the art on ImageNet given by PNASNet [Liu *et al.*, 2018] achieves 82.9% Top1 accuracy.

In this paper, we propose the SuperTML method, which borrows the concept of the Super Characters method to ad-

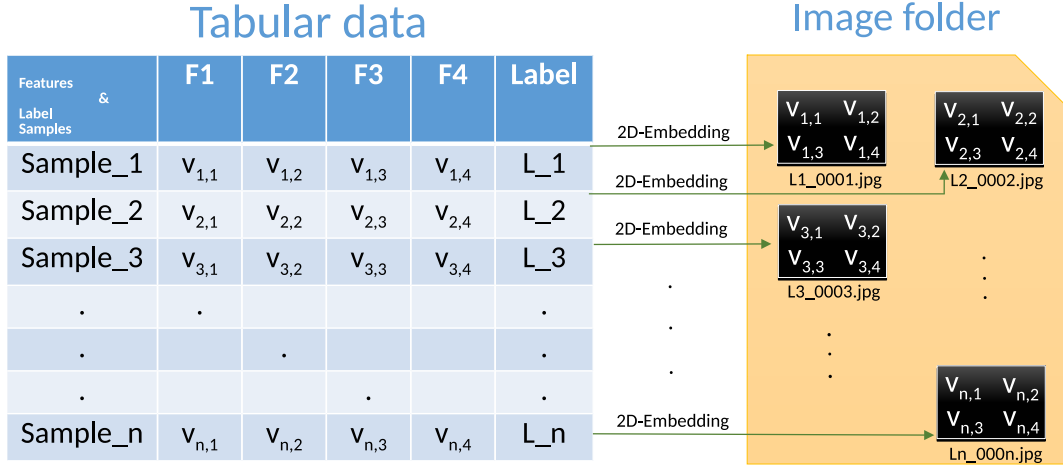


Figure 1: An example of converting training data from tabular into images with two-dimensional embeddings of the features in the tabular data. Therefore, the problem of machine learning for tabular data is converted into image classification problem. The later problem can use pretrained two-dimensional CNN models on ImageNet for finetuning, for example, ResNet, SE-net and PolyNet. The tabular data given in this example has n samples, with each sample having four feature columns, and one label column. For example, assume the tabular data is to predict whether tomorrow’s weather is “Sunny” or “Rainy”. The four features F1, F2, F3, and F4 are respectively “color of the sky”, “Fahrenheit temperature”, “humidity percentage”, and “wind speed in miles per hour”. Sample_1 has class label L1=“Sunny”, with four features values given by $v_{1,1} = \text{“blue”}$, $v_{1,2} = 55$, $v_{1,3} = \text{“missing”}$, and $v_{1,4} = 17$. The two-dimensional embedding of Sample_1 will result in an image of “Sunny_0001.jpg” in the image folder. The four feature values are embedded into the image on different locations of the image. For example, $v_{1,1}$ is a categorical value of color “blue”, so the top left of the image will have exactly the alphabets of “blue” written on it. For another example, $v_{1,2}$ is a numerical value of “23”, so the top right of the image will have exactly the digits of “23” written on it. For yet another example, $v_{1,3}$ should be a numerical value but it is missing in this example, so the bottom left of the image will have exactly the alphabets of “missing” written on it. Other ways of writing the tabular features into image are also possible. For example, “blue” can be written in short as a single letter “b” if it is distinctive to other possible values in its feature column. The image names will be parsed into different classes for image classification. For example, L1 = L2 = “Sunny”, and L3 = Ln = “Rainy”. These will be used as class labels for training in the next step of image classification.

dress problems in the field of TML. For each input, tabular features are first projected into two-dimensional embedding like an image, and this image is fed into finetuned two-dimensional CNN models for classification. The proposed SuperTML method handles the categorical data and missing values in tabular data automatically, without need for conversion into numerical values. Experimental results show that the proposed SuperTML method performs well on both large and small datasets. In one instance of the Higgs Boson Machine Learning Challenge dataset, a single model that applied SuperTML manages to analyze 250,000 training instances and 550,000 testing instances and obtain an AMS score of 3.979, a state-of-the-art result, while the best previous result was 3.806. When using the top three popular databases (ranked by number of clicks since 2007) from UCI Machine Learning Repository, such as the Iris dataset (150 data instances), Adult dataset (48,482 data instances), and Wine dataset (178 data instances), the SuperTML method still achieved state-of-the-art results.

2 The Proposed SuperTML Method

Some considerations of the relationship between TML and the classification task in NLP motivates the proposed SuperTML method. If the tabular features are all treated as tokens as in the NLP tasks, then each sample in the tabular data can be regarded as a concatenation of the tokenized features.

In that case, the current models and methods used in the NLP tasks should be useful for the tabular data as well.

As mentioned in the introduction, Super Characters method using two-dimensional embedding and pretrained CNN models has achieved state-of-the-art result on the text classification tasks. However, different from text classification problems studied in [Sun *et al.*, 2018a], the tabular data has features in separate dimensions. So, the generated images for the tabular data should reserve some gap between features in different dimensions. This will guarantee that the features won’t be overlapped in the generated image.

The proposed SuperTML method addresses the classification problem on tabular data in two steps. The first step is two-dimensional embedding. This step projects features in the tabular data onto the generated images, which is called the SuperTML images in this paper; and the second step is treating the TML problem as an image classification problem, by using finetuned CNN models to classify the generated SuperTML images.

For the first step in the SuperTML method, the conversion of training data from tabular into the SuperTML images is illustrated in Figure 1. It shows an example of classification on the tabular data with four features.

In the second step of the proposed SuperTML method, the recent winning CNN models on ImageNet are used in the experiments in this paper. However, some pretrained models

Algorithm 1 SuperTML_VF: SuperTML method with Variant Fontsize for embedding.

Input: Tabular data training set**Parameter:** Imagesize of the generated SuperTML images**Output:** Finetuned CNN model

- 1: Calculate the feature importance in the given tabular data provided by other machine learning methods.
 - 2: Design the location and fontsize of each feature in order to occupy the imagesize as much as possible. Make sure no overlapping among features.
 - 3: **for** each sample in the tabular data **do**
 - 4: **for** each feature of the sample **do**
 - 5: Draw feature in the designated location and fontsize.
 - 6: **end for**
 - 7: **end for**
 - 8: Finetune the pretrained CNN model on ImageNet with the generated SuperTML images.
 - 9: **return** the trained CNN model on the tabular data
-

are only available to public for specific size of input. For example, the SE-net only published its pretrained model with 224x224 input size; while the PolyNet only published the one with input size of 331x331. These two models are available in the Caffe framework. In order to exclude the accuracy difference brought by different frameworks, NASnet and PNASnet are not used because only TensorFlow models are available.

Figure 1 only shows the generation of SuperTML image for the training data. It is clear that for inference, each testing data should go through the same pre-processing to generate a SuperTML image using the same configuration for two-dimensional embedding, and then be fed into the trained CNN model for classification.

Considering that features may have different importance for the classification task, it is straightforward to allocate larger spaces for features with higher importance and increase the fontsize of the corresponding feature values. The resulting SuperTML_VF method is described in Algorithm 1.

To make the SuperTML more automatic, and remove the dependency on calculating the feature importance as done in Algorithm 1, the SuperTML_EF method is introduced in Algorithm 2. It allocates the same size to every feature, thus tabular data can be directly embedded into SuperTML images without the need of calculation for the feature importance. And this algorithm shows competitive and even better results than 1 as seen later in the experimental section.

3 Experiments

In the experiments below, we use the top three most popular datasets from the UCI Machine Learning Repository [Dua and Karra Taniskidou, 2017] and one wellknown dataset from the Kaggle platform. These four datasets cover a variety of machine learning tasks for tabular data.

The UCI machine learning repository ranks the dataset popularity according their clicks since 2007. Until the date this paper is written, the Iris dataset [Fisher, 1936] is ranked as the most popular dataset with 2.41+ million hits, the Adult dataset [Kohavi and Becker, 1996] (also known as Census

Algorithm 2 SuperTML_EF: SuperTML method with Equal Fontsize for embedding.

Input: Tabular data training set**Parameter:** Imagesize of the generated SuperTML images**Output:** Finetuned CNN model

- 1: **for** each sample in the tabular data **do**
 - 2: **for** each feature of the sample **do**
 - 3: Draw the feature in the same fontsize without overlapping, such that the total features of the sample will occupy the imagesize as much as possible.
 - 4: **end for**
 - 5: **end for**
 - 6: Finetune the pretrained CNN model on ImageNet with the generated SuperTML images.
 - 7: **return** the trained CNN model on the tabular data
-

Salary dataset) ranks the second with 1.40+ million clicks, and the Wine dataset [Forina, 1991] ranks the third with 1.07+ million clicks. Table 1 shows the statistics of these three datasets. The data types of the features covers a variety of integer, categorical, real, and even missing values.

The Kaggle dataset of Higgs Boson Machine Learning Challenge is also used in the experiments. It “attracted an unprecedented number of participants over a short period of time (May 12, 2014 to Sept 15, 2014)” [Adam-Bourdarios *et al.*, 2015]. “There were in total 1,785 teams participated in the competition, one of the largest and most active ones on the platform website www.kaggle.com” [Chen and He, 2015].

For all the three datasets from UCI Machine Learning Repository, SuperTML images of size 224x224 are generated. Pretrained SE-net-154 is finetuned on these three datasets. We also implemented XGBoost and finetune the hyperparameters on each of the three datasets. For Higgs Boson dataset, SuperTML images sizes of 224x224 and 331x331 are both generated for comparison of different pretrained models of SE-net-154 and PolyNet. These two pretrained models have similar Top1 performance on ImageNet (81.32% for SE-net-154, and 81.29% for PolyNet), but different input sizes (224x224 for SE-net-154, and 331x331 for PolyNet).

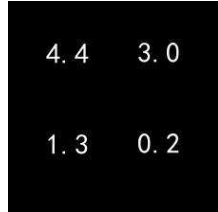
3.1 Experiments on the Iris dataset

“This is perhaps the best known database to be found in the pattern recognition literature” [Fisher, 1936]. The Iris dataset is widely used in machine learning courses and tutorials. It has totally 150 samples with labels of subspecies names, i.e. Iris Setosa, Iris Versicolor, and Iris Virginica. Each sample has four attributes of real numbered features, indicating the measurements of sepal length, sepal width, petal length, and petal width, measured in centimeters.

It is very challenging for applying SuperTML method on this dataset, because this is a very small dataset with only 150 samples. If we split the training and testing by 80%:20%, it means only 120 samples for training, and only 40 samples for each of the class. Deep learning models are data hungry, and the CNN models in computer visions are especially well-known for requiring large amounts of labeled images. For example, ImageNet dataset has over one million images for

Dataset	Classes	#Attributes	Train	Test	Total	Data Types	Missing Values
Iris	3	4	NA	NA	150	Real	No
Wine	3	13	NA	NA	178	Integer and Real	No
Adult	2	14	32,561	12,681	48,842	Integer and Categorical	Yes

Table 1: Datasets statistics used in this paper from UCI Machine Learning Repository. The “NA” in the table denotes that there is no given split for the training and testing dataset.



(a) SuperTML_EF image example for Iris data. Each feature is given equal importance in this example.



(b) SuperTML_VF image example for Wine data. Features are given different sizes according to their importance.

Figure 2: Examples of generated SuperTML image for Iris and Wine dataset.

one thousand classes, with each class having about one thousand labeled samples. When large models are finetuned on this small dataset, there could be a high tendency of overfitting. Furthermore, for this Iris dataset, the data types are all real numbers, which makes the classification using SuperTML method even harder. For methods such as Logistic Regressions, GBT, SVM, and Random Forests, the numerical feature inputs are directly applied to the linear or non-linear models to classify the subspecies. While the CNN models used in the SuperTML method should first learn the shapes of these numerical values of each feature, and then apply non-linear functions on the extracted image features to classify the Iris subspecies. Just to think that the task of recognizing only the shapes of the digits requires quite a lot of data, as shown in MNIST dataset [Deng, 2012].

Figure 2a shows one example of generated SuperTML image for Iris data. The experimental results of using SE-net 154 is shown in Table 2, this result is based on 80%:20% splits of the 150 samples. It shows that the proposed SuperTML method achieves the same accuracy as XGBoost on this small dataset.

Accuracy	Iris(%)	Wine(%)	Adult(%)
xgboost	93.33	96.88	87.32
GB [Biau <i>et al.</i> , 2018]	–	–	86.20
SuperTML	93.33	97.30	87.64

Table 2: Model accuracy comparison on the tabular data from UCI Machine Learning Repository. The splits on Iris and Wine data is 80%:20% as described in the experimental setup.

3.2 Experiments on the Wine dataset

The Wine dataset shares a few similarities to the Iris dataset, so we put this experiment immediately after the Iris experiment. This dataset has the similar task of classifying the input samples into one of the three classes, and the dataset has only 178 samples, which is also a small dataset. The input features are the measurements on Alcohol, Hue, Ash, and etc., in order to predict the type of the wine. In addition, similar to Iris dataset, there is no given split of training and testing datasets in this Wine dataset as well. The number of attributes is 13, which is more than 4 times of that of the Iris dataset. And the data types include not only real valued features, but also integers for feature values. These differences make the classification on Wine data with SuperTML method even harder than in the Iris dataset, because the SuperTML image will have more features to embed, and also a variant size of spaces for each feature due to different data types.

For this dataset, we use SuperTML_VF, which gives features different sizes on the SuperTML image according to their importance score. The feature importance score is obtained using the XGBoost package [Chen and Guestrin, 2016]. One example SuperTML image is shown in Figure 2b. The importance score shows that the feature of Color_intensity is the most important, so we allocate font size of 48 to it in the 224x224 image as seen in the top left corner. Following features of most importance are Flavanoids and Proline, which are allocated on the left following Color_intensity with font size of 48. Similar for the rest of the other features, until the least important features of Proanthocyanins and Nonflavanoid_phenols, as seen on the bottom right corner with font size of 8. The results in Table 2 show that the SuperTML method obtained a little better accuracy than XGBoost on this small dataset.

3.3 Experiments on the Adult dataset

The task of this Adults dataset is to predict whether a person’s income is larger or smaller than 50,000 dollars per year based on the survey data. The survey data have 14 attributes, including age, gender, education, and etc., which contain a mixed type of integer, real valued numbers, and categorical data. It has 32,561 samples for training and 12,681 samples for testing. So, it is convenient to compare with literature results since the training and testing split is given by this dataset. Compared with the other two datasets from UCI Machine Learning Repository, this relative large dataset is in favour of deep learning models to be used in SuperTML method.

For categorical features which are given by strings of text, the Squared English Word (SEW) method [Sun *et al.*, 2019] is used. The benefits of writing English word in this format is

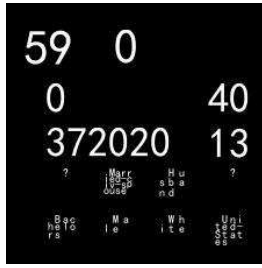


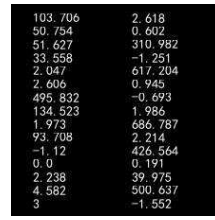
Figure 3: SuperTML image example from Adult dataset. This sample has yearly salary larger than 50k with its features given different sizes in the SuperTML image according to their importance given by third party softwares. To make this SuperTML method to automatically generate two-dimensional images without the dependencies on third party softwares, generating SuperTML images without considerations of feature importance is also experimented in section 3.4. This sample has age=59, capital gain = 0, capital loss=0, hours per week = 40, fnlweight = 372020, education number = 13, occupation = "???" (missing value as given in the data, it can also be replaced by "MissingValue" in the SuperTML image), marital status = "Married-civ-spouse", relationship = "Husband", workclass = "???" (missing value), education="Bachelors", sex = "Male", race = "White", native country = "United-States".

two-folded. Firstly, Super Characters method has shown state of the art performance on Asian languages, such as Chinese, Japanese, and Korean, which has their characters written in the form of glyphs in a squared space. Motivated by this, writing English word in SEW format converts the and guarantees that each word will be written in a unique way to distinguish from other words. Secondly, writing the features expressed by English strings in this format gurantees that each feature occupies the same position without any change caused by the length of the feature string.

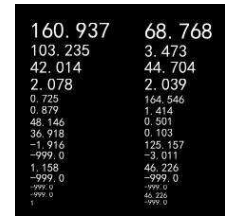
One example of generated SuperTML image is given in Figure 3. Table 2 shows the results on Adult dataset. On this dataset, Biau and et. al. [2018] proposed Accelerated Gradient Boosting (AGB) and compared the performance with original Gradient Boosting (GB) with series of finetuned hyperparameters. The best accuracy is given by GB when the shrinkage parameter is set at 0.1. We also implemented XGBoost on this dataset and preprocessed the categorical data by using integer encoding (in python pandas, using `astype('category')`). The XGBoost finally gives the best result of 87.32 after finetuning the number of trees at 48. We can see that on this dataset, the proposed SuperTML method still beats finetuned XGBoost by 0.32% accuracy.

3.4 Experiments on the Higgs Boson Machine Learning Challenge dataset

The Higgs Boson Machine Learning Challenge is a binary classification task to classify quantum events as signal or background. It was hosted by Kaggle and now the challenge data is available on opendata [Adam-Bourdarios *et al.*, 2015]. It has 25,000 examples for training, and 55,000 examples for testing. Each example has 30 features, which are all real-valued. In this challenge, AMS score [Adam-Bourdarios *et al.*, 2015] is used in the Kaggle contest as the performance metric.



(a) SuperTML image example for Higgs Boson data. Each feature is given equal importance in this example.



(b) SuperTML image example for Higgs Boson data. Features are given different sizes according to their importance.

Figure 4: Examples of generated SuperTML image for Higgs Boson dataset.

The reason why we select this dataset is two folded. First, it is a well-known dataset and successful models such as xgboost and Regularized Greedy Forest are used in this dataset. Second, the performance metric used in this dataset is AMS score instead of accuracy. It will be interesting and challenging to compare the performance of SuperTML method using a different metric with other methods on this well known dataset for tabular data.

The SuperTML images of size 224x224 are generated for finetuning the SE-net models, and the images of size 331x331 are generated for finetuning the PolyNet models. Figure 4a shows an example of example of "background" event, which is generated into an SuperTML image of 224x224, with equal space for every feature value in the tabular data. Figure 4b shows an example of "signal" event, with variant sizes for features according to their importance, also in an 224x224 image. The 331x331 SuperTML images are similar to 224x224 images except that each feature is scaled up in the SuperTML image.

As pointed by [Chen and Guestrin, 2016] that the AMS is an unstable measure, and AMS is not chosen as a direct objective for XGBoost implementation in this challenge. For simplicity, softmax loss is still used in this dataset as objective to minimize. Table 3 shows the comparison of different algorithms. The DNN method and neural networks used in the first and third rows are using the numerical values of the features as input to the models, which is different from SuperTML method using two-dimensional embeddings. It shows that SuperTML method with equal fontsize gives the best AMS score of 3.979. The PolyNet models trained with larger size of 331x331 does not help improve the AMS score. In addition, the SuperTML_EF seems even better than SuperTML_VF results for both 224x224 and 331x331 image sizes, which indicates SuperTML method can work well without the calculation of the importance scores.

3.5 Error Analysis

The experiment using the SuperTML method on the Iris dataset with 80%:20% split for training and testing, has 2 testing samples giving the wrong prediction. We take one of the wrong prediction sample in the testing dataset for error analysis. Its ground truth label is Iris-virginica, but was incorrectly

Methods	AMS
DNN by Gabor Meli	3.806
RGF and meta ensemble by Tim Salimans	3.789
Ensemble of neural networks by nhx5haze	3.787
xgboost	3.761
SuperTML_EF(224x224)	3.979
SuperTML_VF (224x224)	3.838
SuperTML_EF (331x331)	3.934
SuperTML_VF (331x331)	3.812

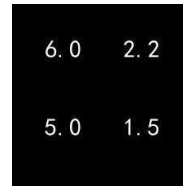
Table 3: Comparison of AMS score on Higgs Boson dataset for different methods. The first four rows are top rankers in the leaderboard in the Higgs Boson Challenge.

classified as Iris-versicolor. Its four features are 6.0, 2.2, 5.0, and 1.5 respectively, as shown in Figure 5a. All the training samples with common integer portion for each of the four feature values, namely 6, 2, 5, and 1 are taken into comparison in Figure 5b-Figure 5f. But these samples may have different decimal values. It shows that this SuperTML image of this virginica example in Figure 5a looks more like the versicolor sample in Figure 5b than the other virginica samples in Figure 5c-Figure 5f, when the shape of numbers in decimal portion is compared. So, the testing sample of virginica in Figure 5a is more likely to be classified as versicolor, which is the label for 5b.

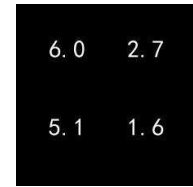
The features in this Iris dataset are all numerical values without missing numbers. During model training, these SuperTML images of numerical features are fed into the two-dimensional CNN model which classifies images based on the pixel values and the relationship between pixels. At inference time, the model classifies the samples based on the “appearance” of the features in real-valued numbers. But the numerical values has some hidden relationship behind the shape of the digits which are hard for the CNN model to learn, such as 6.01 and 5.999 which are both approximate to the number of 6.00, even though their shape is not alike.

4 Conclusion and Future Work

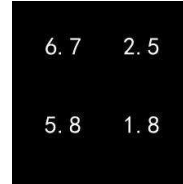
The proposed SuperTML method borrows the idea from Super Characters and two-dimensional embedding, and fine-tunes the pre-trained CNN models on unstructured data for transfer learning the structured data in the tabular form. Experimental results shows that the proposed SuperTML method has achieved state-of-the-art results on both large and small tabular dataset. As some low power domain specific CNN accelerators [Sun *et al.*, 2018b] are available in the market, the SuperTML method has huge potentials in practical applications in the real world. For example, for IoT (Internet of Things) applications in smart home scenarios, current machine learning solutions at the edge are still using Logistic Regression models which is computationally inexpensive but is expected to be less accurate compared with large models like the CNN models used in the SuperTML method. Using these low-power CNN accelerators with the SuperTML method, it is possible to provide low-power and high accuracy models at the edge devices. The future work could go



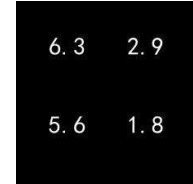
(a) The test SuperTML image of an virginica sample.



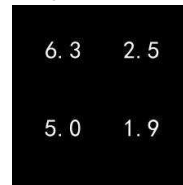
(b) One example from training set with label “versicolor”.



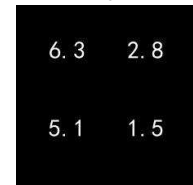
(c) First example from training set with label “virginica”.



(d) Second example from training set with label “virginica”.



(e) Third example from training set with label “virginica”.



(f) Fourth example from training set with label “virginica”.

Figure 5: Error analysis on an Iris-virginica input wrongly predicted as Iris-versicolor. These six samples have the common integer for each of the four feature values, namely 6, 2, 5, and 1. But these samples may have different decimal values. Figure 5c-Figure 5f are the only four training samples with the integer portion of the feature values same as Figure 5a. But one “versicolor” sample from training set as shown in Figure 5b not only has the same integer part as Figure 5a, but also has more similar shapes of decimal part for each of the four feature values to Figure 5a than the other four samples. The CNN models that learn the classification model based on the shape of the feature values written on the image have high tendency to classify the example in Figure 5a as the same category of Figure 5b.

four directions. First, given the success of Super Characters method [Sun *et al.*, 2018a] in text classification, categorical type of data, and also dataset with missing values, should be advantage for SuperTML method. Unlike the numerical features, the categorical feature has all the information on the literal category names written in the text. Second, compare with not only the Gradient Boosting method, but also the one-dimensional embedding based RNN and CNN methods on the TML tasks. Third, some more powerful CNN models could be used in SuperTML method, such as NASNet, PNASnet, and the other successful models from architecture search. Fourth, modify the model architectures to add attention scheme in order to take into account of variant feature importance.

References

- [Adam-Bourdarios *et al.*, 2015] Claire Adam-Bourdarios, Glen Cowan, Cécile Germain, Isabelle Guyon, Balázs Kégl, and David Rousseau. The higgs boson machine learning challenge. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pages 19–55, 2015.
- [Biau *et al.*, 2018] Gérard Biau, Benoît Cadre, and Laurent Rouvière. Accelerated gradient boosting. *Machine Learning*, pages 1–22, 2018.
- [Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.
- [Chen and He, 2015] Tianqi Chen and Tong He. Higgs boson discovery with boosted trees. In *NIPS 2014 workshop on high-energy physics and machine learning*, pages 69–80, 2015.
- [Chen *et al.*, 2016] Ting Chen, Lu-An Tang, Yizhou Sun, Zhengzhang Chen, and Kai Zhang. Entity embedding-based anomaly detection for heterogeneous categorical events. In *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1396–1403, 2016.
- [Deng, 2012] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [Dua and Karra Taniskidou, 2017] Dheeru Dua and Efi Karra Taniskidou. UCI machine learning repository, 2017.
- [Fisher, 1936] R.A. Fisher. Iris data set from UCI machine learning repository, 1936.
- [Fogg, 2016] Andrew Fogg. Anthony goldbloom gives you the secret to winning kaggle competitions, 2016.
- [Forina, 1991] PARVUS Forina, M. et al. Wine data set from UCI machine learning repository, 1991.
- [Guo and Berkhahn, 2016] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [Hu *et al.*, 2018] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.
- [Kohavi and Becker, 1996] Ronny Kohavi and Barry Becker. Adult data set from UCI machine learning repository, 1996.
- [Lam *et al.*, 2018] Hoang Thanh Lam, Tran Ngoc Minh, Mathieu Sinn, Beat Buesser, and Martin Wistuba. Neural feature learning from relational database. *arXiv preprint arXiv:1801.05372*, 2018.
- [Liu *et al.*, 2018] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.
- [Prokhorenkova *et al.*, 2018] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pages 6639–6649, 2018.
- [Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [Sun *et al.*, 2018a] Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. Super characters: A conversion from sentiment classification to image classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 309–315, 2018.
- [Sun *et al.*, 2018b] Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. Ultra power-efficient cnn domain specific accelerator with 9.3 tops/watt for mobile and embedded applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1677–1685, 2018.
- [Sun *et al.*, 2019] Baohua Sun, Lin Yang, Catherine Chi, Wenhan Zhang, and Michael Lin. Squared english word: A method of generating glyph to use super characters for sentiment analysis. *arXiv preprint arXiv:1902.02160*, 2019.
- [Thomas, 2018] Rachel Thomas. An introduction to deep learning for tabular data, 2018.
- [Vorhies, 2016] William Vorhies. Has deep learning made traditional machine learning irrelevant?, 2016.
- [Zhang *et al.*, 2017] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 718–726, 2017.
- [Zoph *et al.*, 2018] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.