Prediction of processing defect of CNC machine

Lee hyo jeong
AIGS
UNIST
macarize@unist.ac.kr

Abstract

In CNC process, It is crucial to predict the defects according to tool life. Previously, manufacturers could not detect the defection of products before the whole production process is over. Therefore, We intended to predict processing defect of computerized numerical control(CNC) machine using various machine data. This way, we can detect defections earlier and save time and costs. We achieve this by designing model composed of several fully connected layers. And we also included the experiment using supervised contrastive loss in shorter training scenarios.

1 Introduction

In the old general machine tool (shelf, milling), the movement of the tool is achieved by manual handle operation. The CNC machine tool automatically controls and processes the movement by the CNC(Computerized Numeric Control) program. In CNC processing, the machine performs NC work and computer programs are customized for objects and use CNC processing language (G-code). This machine is programmed to provide basic functionality such as transfer speed, coordinates, position, and speed be controlled by. So CNC machining allows you to pinpoint the location.

In CNC cutting, the cutting tool is worn out by friction or the instantaneous change of cutting force As a result, when the wear limit is reached due to damage, the processing accuracy decreases rapidly. In addition, the quality of the manufactured product as it affects the life of the machine tool due to the increase in cutting power, In order to make it uniform and improve reproducibility, the prediction of processing defects according to tool life is required.

Previously, manufacturers could not detect the defection of products before the whole production process is over. Therefore, We intended to predict processing defect of computerized numerical control(CNC) machine using various machine data. This way, we can detect defections earlier and save time and costs. We achieve this by designing model composed of several fully connected layers. And we also included the experiment using supervised contrastive loss in shorter training scenarios.

2 Related Works

2.1 Anomaly detection using deep learning

Deep hybrid models for anomaly detection use deep neural networks mainly autoencoders as feature extractors, the features learned within the hidden representations of autoencoders are input to traditional anomaly detection algorithms such as one-class SVM (OC-SVM) to detect outliers [Andrews et al. (2016)]. Figure 7 illustrates the deep hybrid model architecture used for anomaly detection. Following the success of transfer learning to obtain rich representative features from models pre-trained on large data-sets, hybrid models have also employed these pre-trained transfer learning models as feature extractors with great success. A notable shortcoming of these hybrid approaches is the lack of trainable objective customized for anomaly detection, hence these models fail to extract

Preprint. Under review.

rich differential features to detect outliers. In order to overcome this limitation customized objective for anomaly detection such as Deep one-class classification [?] and One class neural networks [Chalapathy et al. (2018)] is introduced.

One class neural network (OC-NN) [Chalapathy et al. (2018)] methods are inspired by kernel-based one-class classification which combines the ability of deep networks to extract a progressively rich representation of data with the one-class objective of creating a tight envelope around normal data. The OC-NN approach breaks new ground for the following crucial reason: data representation in the hidden layer is driven by the OC-NN objective and is thus customized for anomaly detection. This is a departure from other approaches which use a hybrid approach of learning deep features using an autoencoder and then feeding the features into a separate anomaly detection method like one-class SVM (OC-SVM). For baseline model, we followed similar scheme as OC-NN architecture.

2.2 Contrastive learning

Contrastive learning has recently received interest due to its success in self-supervised representation learning. Inspired by the learning strategy that humans utilize when given a few examples, key idea of contrastive learning can be considered as learning by comparing. Constrative loss[Chopra et al. (2005)] is one of the earliest family of constrative loss which only one positive and one negative sample are involved. N-pair loss [Chopra et al. (2005)] generalizes contrastive loss to include comparison with multiple negative samples. Soft nearest neighbor loss [Frosst et al. (2019)] included multiple positive samples to be participated. For the supervised setting where labels are known, supervised contrastive loss [Khosla et al. (2020)] is proposed, aiming to leverage label information more effectively than cross entropy, imposing that normalized embeddins from the same class are closer together than embeddings from different classes. Contrastive representation learning is well exploited in computer vision domain like generating class-agnostic activation map for weakly supervised object localization and semantic segmentation [Xie et al. (2022)], Image dehazaing [Wu et al. (2021)]

3 Dataset

Dataset for this experiment is compromised of various machine data in regards to CNC machine. Total number of train examples are 18806 and number of test examples are 13242. Number of attributes are 48 and number of labels are 2 that indicates whether the product is defected or not. Given that the labels are mutually exclusive, we simplyfied the ground truth vector to have length of one binary indicator.

4 Methods

4.1 Preprocessing

In this step, we vectorized string type attributes and scaled the values.

4.2 Baseline model and training

The model is consist of sequence of fully connected layers. Detailed structure is as follows : FC(48 x 128, relu), FC(128 x 256, relu), FC(256 x 512, relu) x 2, FC(512 X 256, relu), FC(256 x 128, relu), FC(128 X 1, sigmoid). Dropout layers[Srivastava et al. (2014)] follows after each fully connected layers with probability 0.3. Used optimizer is Adam[Kingma and Ba (2014)]. And used binary cross entropy for criterion. Batch size is 1024 and fixed learning rate is 1e-4.

4.3 Baseline model with contrastive loss

We constructive consecutive experiment with supervised contrastive loss[Khosla et al. (2020)]. Given a set of randomly sampled n (image, label) pairs, $x_i, y_{i}{}_{i=1}^{n}, 2n$ training pairs can be created by applying two random augmentation of every sample, $\tilde{X}_I, \tilde{y}_i{}^2 n_(i = 1)$

$$L_{supcon} = -\sum_{i=1}^{2n} \frac{1}{2|N_i| - 1} \sum_{j \in N(yi), j \neq i} log \frac{exp(z_i \cdot z_j / \tau)}{\sum_{k \in I, k \neq i} exp(z_i \cdot z_k / \tau)}$$

Where $z_k = P(E(\tilde{x}_k))$, in which $E(\cdot)$ is and encoder network, $P(\cdot)$ is a projection network. $N_i = j \in I : \tilde{y}_j = \tilde{y}_i$ contains a set of indices of samples with label y_i. Including more positive samples into the set N_i leads to improved results. In this experiments, we constructed positive pairs with samples with same labels, not augmented self. For the final loss, we used sum of supervised contrastive loss and cross entropy loss.

5 Results

Following tables are comparison between baseline model and baseline model with supervised contrastive loss. First table is the case when training model with epoch 300 and second table is the case when training model with epoch 5. Model with supervised contrastive loss shows slightly better performance when training the full epoch. With fewer epoch, model with unsupervised contrastive loss shows noticeably better performance. But found it quite unstable compared to model with binary cross entropy loss only.

Color	Baseline	Baseline SCL
Acc	91.42	91.56

Table 1: Training with epoch 300.

Color	Baseline	Baseline SCL
Acc	78.19	82.78

Table 2: Training with epoch 5.

References

Andrews, J. T., Morton, E. J., and Griffin, L. D. (2016). Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, 6(1):21.

Chalapathy, R., Menon, A. K., and Chawla, S. (2018). Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1.

Frosst, N., Papernot, N., and Hinton, G. (2019). Analyzing and improving representations with the soft nearest neighbor loss.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Wu, H., Qu, Y., Lin, S., Zhou, J., Qiao, R., Zhang, Z., Xie, Y., and Ma, L. (2021). Contrastive learning for compact single image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10551–10560.

Xie, J., Xiang, J., Chen, J., Hou, X., Zhao, X., and Shen, L. (2022). Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation.