# Classifying Tweets using Convolutional Neural Networks with Multi-Channel Distributed Representation

Shuichi Hashida, Keiichi Tamura, Tatsuhiro Sakai

*Abstract*—**This paper is focused on a state-of-art classification method for short text messages. With the increasing interest in social media, people are posting many short text messages, not only to communicate with other people, but also to share information. This trend has led to a new research area, microblog mining. This type of data mining can extract real-world topics and events from microblogs. However, because text messages on a micro-blogging site are short, their classification is a challenging task. In particular, Twitter is one of the most well-known micro-blogging services, where tweets frequently users' reactions to real-world topics and events occurring in their surroundings. In our previous paper, we proposed a real-time topic monitoring system that implements a naive Bayes classifier to classify tweets into two classes: "relevant" and "irrelevant" to a monitored topic. The classification performance has limitations, because the naive Bayes' classification is based on word probabilities. To address this problem, we propose a deep-learning-based classification method that features a new distributed representation for words, multi-channel distributed representation. Distributed representation indicates word vectors representing the latent features of words. To enhance the capability of a distributed representation, each of its items has several channel values in a multi-channel distributed representation. In our experiments, we evaluated our model's performance in comparison with that of other convolutional neural network (CNN) models and a long short-term memory model. The results showed that the classification performance of the deep learning models was superior to that of the naive Bayes classifier. Moreover, a CNN with multi-channel distributed representation can classify tweets better than a CNN without multi-channel distributed representation.**

*Index Terms*—**Deep learning, Distributed representation, Text classification, CNN.**

## I. Introduction

COMMUNICATION through social media has an important role in our daily life, and therefore, people are extremely active in sharing their life experience by posting messages [1]. Users of Twitter, which is one of the most influential information sources, tweet about various real-world topics and events, including tourist spots, local events, natural disasters, accidents and news. The possibility of extracting useful information from tweets has attracted the attention of a very large number of researchers [2], [3]. This trend has opened a new research area that includes topic and event detection, trend analysis, and marketing using tweets in many different application domains [4], [5], [6].

We developed a test bed for a topic analysis system that can observe the dynamics of real-world topics [7]. To extract tweets that are related to a topic being monitored in this system, a naive Byes classifier [8] is used for classifying tweets. Tweets are classified according to whether their content is or is not related to the topic being monitored. A natural disaster observation system that can detect areas suffering from heavy rain or snow and the time periods of the precipitation by using the topic analysis system was implemented. This system significantly affects the analysis of real-world information using tweets posted on the Twitter site; however, the quality of the information input to the system depends on the classification performance. As the classifier is based on naive Bayes, there is room for improving the classification performance.

In this paper, we propose a new classifier that is based on a deep learning technique to improve the classification performance of the topic analysis system and present a detailed evaluation that includes a comparison of our model with several methods. The proposed classifier utilizes a convolutional neural network (CNN), where the input to the network is a sequence of multi-channel distributed representations. The model is based on a CNN-based text classifier that was proposed by Kim [9]. The key idea of their model is that the size of the convolution filters is the same as the number of dimensions of the distributed representation. A text sentence comprises a time series sequence of words, and multi-filters are used to capture the features of sentences. These techniques can maintain the word information through the convolution processes and implement a CNN to classify text data.

The efficacy of Kim's model depends on the capability of the input representation. To enhance the capability of the distributed representation, we developed multi-channel distributed representation. A text message in a tweet is a sequence of words and is referred to as a sequence of distributed representations. In the proposed model, this sequence is mapped to multiple sequences on time delay, and each sequence is a channel. To evaluate the proposed model, in our experiments, we used actual tweets, classifying them according to their relevance for an actual social media topic. Moreover, we compared the performance of the proposed model [10] with that of several deep-learning based classifier methods and Kim's method. The results showed that the classification performance of the deep learning models was superior to the naive Bayes method. Moreover, a CNN model with multi-channel distributed representation can classify tweets better than a CNN without multi-channel distributed representation.

The rest of the paper is organized as follows. Sections II and III provide an overview of related work and our previous work. Section IV provides an explanation of the proposed model. In Section V, the experimental results are reported. Finally, in Section VI we conclude the paper and present our future work.

S. Hashida and K. Tamura are with Graduate School of Information Sciences, Hiroshima City University, 3-4-1, Ozuka-Higashi, Asa-Minami-Ku, Hiroshima 731-3194 Japan; corresponding e-mail of corresponding author: (ktamura@hiroshima-cu.ac.jp).

T. Sakai is Japan Society for the Promotion of Science Research Fellow (PD).

## II. Related Work

This section provides an overview of the related work on identifying tweets for topic and event extraction by means of machine learning, sentiment identification, and text data classification using deep learning techniques. Social media constitute a new type of information source, which shows very rapid growth. In particular, people obtain information instantaneously about trending topics and events from tweets on Twitter, which is one of the most widely used microblogging services [5]. A large number of tweets include only inconsequential information; the extraction of tweets involving useful information is important for the use of tweets in many different domains.

A survey and a comparative study of tweet sentiment analysis were reported by Silva et al. [11]. In tweet sentiment analysis based on machine learning, an emotion of a tweet is identified by means of a classifier. Davidov et al. [12] identified tweet polarity by using emoticons as class labels. Their method defines the feature vector of a tweet and it by using a $k$-nearest neighbor algorithm. Aramaki et al. [13] proposed a novel method for detecting influenza epidemics by means of tweets; their method utilizes classifiers, such as a support vector machine (SVM) and naive Bayes, to extract tweets that include topics about influenza. In our previous study [7], a classifier that utilizes a Bayes technique was used to classify tweets.

Classifiers embedding neural network techniques have been studied for application to text data. These types of classifiers have gained renewed attention owing to the recent development of deep learning techniques. Kim [9] proposed a deep-learning-based sentence classifier that uses a CNN. Kim utilized deep learning with distributed representation to classify sentences. Severyn et al. [14] proposed a deep-learning-based tweet classifier that uses Kim's model. There are several methods for expressing the features of words in a sentence, including one-hot vector and distributed representation. In Kim's model, distributed representation is used as the word vector and a sequence of word vectors are input to the network. Georgakopoulos et al. [15] proposed a toxic comment classification method that uses a CNN. Rios et al. [16] also proposed a biomedical text classification method that uses a CNN.

Many methods exist for expressing the features of words in a sentence through neural language models [17], [18]. In neural language processing, the deep model learns the word vector to decrease the dimension of the word expression. In contrast, in [19], microblog texts were mapped to low dimensional vector spaces by means of deep learning. Shuang et al. [20] considered word order, which is important for sentence sentiment classification, designing an encode-decode model called convolutional neural network long-short-term memory (CNN-LSTM). In this study, we focused on feature expression methods for words in a sentence. For incorporating time delay information into the distributed representation of a word, a multi-channel distributed representation is proposed.

## III. Topic Analysis System based on Density-based Spatiotemporal Clustering

In this section, we explain our previous method, which is a topic analysis system based on density-based spatiotemporal clustering [7]. Fig. 1 shows an overview of the topic analysis system. The system has three main stages: tweet classification, spatiotemporal clustering, and visualization through a Web application. Users set a monitoring, topic such as heavy rain and snow. The tweet classifier classifies tweets according to whether or not their content is related to the monitoring topic. If the tweet classifier achieves a high performance level, the effectiveness of the system is improved. In the spatiotemporal clustering, areas with high densities of tweets constituting spatiotemporal clusters are extracted by density-based spatiotemporal clustering. Spatiotemporal clusters are extracted in real time by processing every geo-tagged tweet that arrives.

The process flow of the system is as follows.

1) First, the system crawls geo-tagged tweets from Twitter using the Geo-tagged Tweet Crawler. The geo-tagged tweets are stored in the Geo-tagged Tweet Database.
2) Next, the tweet classifier, which is based on naive Bayes, classifies the geo-tagged tweets into relevant or irrelevant geo-tagged tweets. The relevant geo-tagged tweets are passed to the Spatiotemporal Clustering stage.
3) Then, the Spatiotemporal Clustering stage utilizes $(\epsilon, \tau)$-density-based adaptive spatiotemporal clustering [21] to extract spatiotemporal clusters as areas related to the topic being monitored.
4) Finally, the Web application visualizes the tweet clusters on a map.

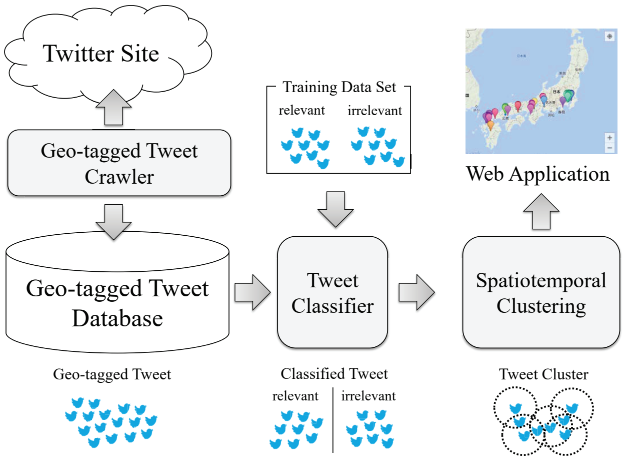
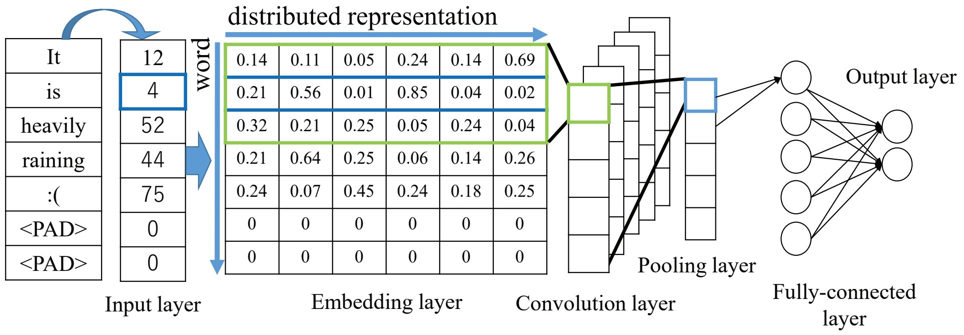


Fig. 1. Overview of the topic analysis system

## IV. Text Classification usng Convolutional Neural Network Model

Our proposed model was inspired by Kim's model, which is based on CNNs [22]. CNNs were originally proposed for image classification, but can be applied to various types of data, such as text data, time series numeric data, and multimodal data. A CNN is a multiple-layer neural network that consists of convolutional layers, pooling layers, and fully connected layers as hidden layers. In a convolution layer, a filter is slid across the data, and feature maps are extracted through element-wise multiplication. In general, a convolution layer has multiple filters, where one filter creates one feature map. The role of this layer is feature extraction.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| It | 12 | 0.14 | 0.11 | 0.05 | 0.24 | 0.14 | 0.69 |
| is | 4 | 0.21 | 0.56 | 0.01 | 0.85 | 0.04 | 0.02 |
| heavily | 52 | 0.32 | 0.21 | 0.25 | 0.05 | 0.24 | 0.04 |
| raining | 44 | 0.21 | 0.64 | 0.25 | 0.06 | 0.14 | 0.26 |
| :( | 75 | 0.24 | 0.07 | 0.45 | 0.24 | 0.18 | 0.25 |
| <PAD> | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| <PAD> | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Fig. 2. Kim's model. There are six main structures, an input layer, embedding layer, convolution layer, pooling layer, fully-connected layer, and output layer.

A pooling layer is a type of down sampling layer. It is responsible for reducing the spatial size of the feature maps' and its role is to abstract information. A fully connected layer is the same as an affine layer in a multi-layer perceptron neural network. This layer usually represents the middle- and high-level features of input data.

In Kim's model, there are six main structures, an input layer, embedding layer, convolution layer, pooling layer, fully-connected layer, and output layer (Fig. 2). The input layer of Kim's model accepts a sequence of word identifications of the text data. The first layer is an embedding layer that converts each identification in the input to a multi-dimensional vector. In the embedding layer, the sequence of word identifications is converted to a two-dimensional array, where the $i$-th data in a row represents the $i$-th word in the text data and the $j$-th column data represents the $j$-th attribute of distributed representation. The third layer conducts convolutions over the multi-dimensional vectors by using multiple filter sizes. To maintain word information, the horizontal filter size is the same length as that of a word vector. The fourth layer is a max-pooling layer that max-pools the result of the convolution-layer into a one-feature vector. The max-pooling layer connects to the fully-connected layer, which in turn connects to the softmax output layer.

Distributed representation is a word embedding technique that extracts the features of words in a multi-dimensional feature space. In natural language processing, words are usually represented as identification numbers having no meaning (e.g., "rain"→1 and "snow"→2). Distributed representation attempts to map words represented by identification numbers to $d$-dimensional vectors in a continuous vector space such that similar types of words are mapped to the same space (Fig. 3). The utilization of distributed representation allows a model to learn the features of text referred to as a sequence of words, because similar words are represented as similar vectors.

Pre-trained and online-trained models are used to create distributed representation. In pre-trained models, distributed representation is extracted using a model trained on a large-scale text dataset. In the study in [9], distributed representation was created by using an unsupervised neural language model. The authors used *word2vec* as distributed representation extracted from a learned model trained on 100 billion words from Google News [23]. In online-trained models, distributed representation is extracted using embedded layers learned by user-given training datasets, and the entire neural network is learned by the user-given training datasets. In this study, in the proposed model, the online-trained model was utilized. Therefore, distributed representation was extracted using an embedded layer learned by user-given training tweet datasets.

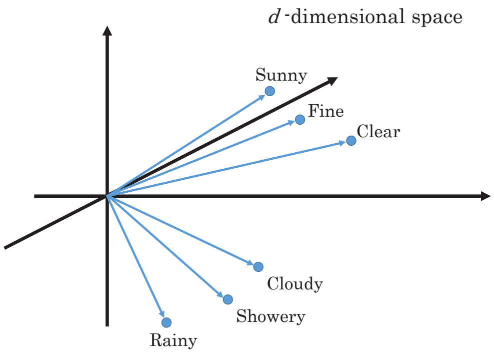
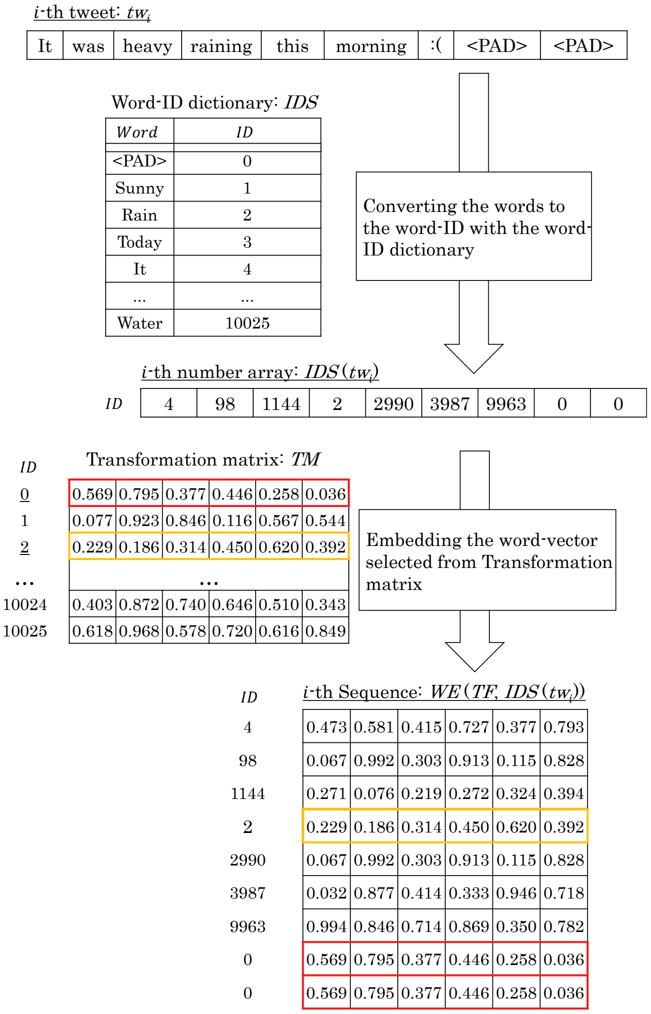


Fig. 3. Mapping to $d$-dimensional space

## V. Proposed Model

The proposed model consists of six main structures, an input layer, embedding layer, convolution layer, pooling layer, fully-connected layer, and output layer. The input layer of the model accepts a sequence of word identifications of the text data. The embedding layer outputs the $d$-dimensional vectors of inputs represented by a sequence of identification numbers mapped to words. A sequence of the $d$-dimensional vectors is referred to as a two-dimensional array. The embedding layer has a transformation matrix $TF \in \mathcal{R}^{N \times d}$, where the total number of words is $N$ and $d$ is the dimension number of the distributed representations. The $i$-th row of the transformation matrix corresponds to the identification number $i$. $tf_{i,j}$ in $TF$ shows the $j$-th dimensional value for the word mapping to identification number $i$. For example, let the identification number of the word "cloudy" be 100. The word vector of "cloudy" is $TF_{100} = (tf_{100,1}, tf_{100,2}, \cdots, tf_{100,d})$.

Fig. 4 illustrates a simple algorithm in the embedding layer. Text data in a tweet $tw_i$ is a sequence of words $tw_i =< word_{i,1}, word_{i,2}, \cdots, word_{i,m} >$, where $m$ is the length of the word vector or distributed representation; moreover, it is a sequence of identification numbers or a word identification array $IDS(tw_i) =< ids_{i,1}, ids_{i,2}, \cdots, ids_{i,m} >$. In the embedding layer, the word vector corresponding to each identification number is searched in $TF$. The embedding layer outputs a sequence of vectors $WE(TF, IDS(tw_i)) =< TF_{ids_{i,1}}, TF_{ids_{i,2}}, \cdots, TF_{ids_{i,m}} >$. The default value of $tf_{i,j}$ is set to a random value. The value of $tf_{i,j}$ is revised iteratively by means of a training process.



*i*-th tweet: $tw_i$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| It | was | heavy | raining | this | morning | :( | <PAD> | <PAD> |

Word-ID dictionary: *IDS*

| *Word* | *ID* |
|---|---|
| <PAD> | 0 |
| Sunny | 1 |
| Rain | 2 |
| Today | 3 |
| It | 4 |
| ... | ... |
| Water | 10025 |

Converting the words to the word-ID with the word-ID dictionary

*i*-th number array: $IDS(tw_i)$

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| *ID* | 4 | 98 | 1144 | 2 | 2990 | 3987 | 9963 | 0 | 0 |

Transformation matrix: *TM*

| *ID* | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0.569 | 0.795 | 0.377 | 0.446 | 0.258 | 0.036 |
| 1 | 0.077 | 0.923 | 0.846 | 0.116 | 0.567 | 0.544 |
| 2 | 0.229 | 0.186 | 0.314 | 0.450 | 0.620 | 0.392 |
| ... | ... | | | | | |
| 10024 | 0.403 | 0.872 | 0.740 | 0.646 | 0.510 | 0.343 |
| 10025 | 0.618 | 0.968 | 0.578 | 0.720 | 0.616 | 0.849 |

Embedding the word-vector selected from Transformation matrix

*i*-th Sequence: $WE(TF, IDS(tw_i))$

| *ID* | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 0.473 | 0.581 | 0.415 | 0.727 | 0.377 | 0.793 |
| 98 | 0.067 | 0.992 | 0.303 | 0.913 | 0.115 | 0.828 |
| 1144 | 0.271 | 0.076 | 0.219 | 0.272 | 0.324 | 0.394 |
| 2 | 0.229 | 0.186 | 0.314 | 0.450 | 0.620 | 0.392 |
| 2990 | 0.067 | 0.992 | 0.303 | 0.913 | 0.115 | 0.828 |
| 3987 | 0.032 | 0.877 | 0.414 | 0.333 | 0.946 | 0.718 |
| 9963 | 0.994 | 0.846 | 0.714 | 0.869 | 0.350 | 0.782 |
| 0 | 0.569 | 0.795 | 0.377 | 0.446 | 0.258 | 0.036 |
| 0 | 0.569 | 0.795 | 0.377 | 0.446 | 0.258 | 0.036 |

Fig. 4. Embedding layer

In the proposed model, distributed representation of words can be extracted through an embedding layer. Distributed representation is a high-level representation representing the latent feature of a word in a training dataset; however, it cannot capture sentence structure. For example, suppose that there are two tweets: “It is snowy today” and “It will be snowy tomorrow.” Although the contexts of the words “snowy” convey different meaning, the word vectors of both instances of “snowy” are the same. This causes degradation model’s performance. To consider the context of a word, in this paper, multi-channel distributed representation is proposed.

In multi-channel distributed representation, each element of the vector of a word has multiple values representing multi-channels. Sequences of word vectors for these time delay sequences are obtained through embedded layers. Sequences of word vectors are stacked; therefore, each word has different word vectors. Although the same word is present in various tweets, if the context of tweets is different, the word is mapped to different multi-channeled word vectors.

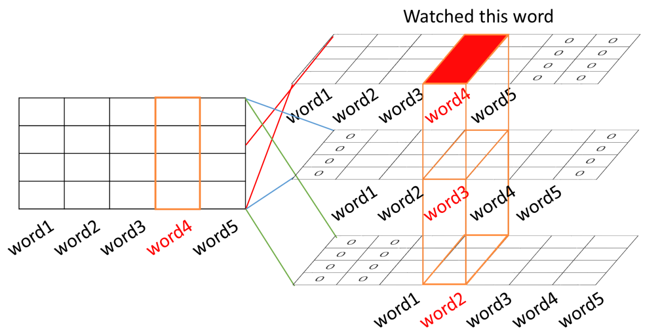


Fig. 5. Conversion distributed representation to multi-channel

Suppose that $k$ represents the number of channels. Then, the processing of generating multi-channel distributed representation is as follows.

1) A sequence of identification numbers is created from a tweet $tw_i$. Let a sequence of identification numbers be $IDS(tw_i) =< ids_{i,1}, ids_{i,2}, \cdots, ids_{i,m} >$.
2) $IDS(tw_i)$ is converted to a sequence of word vectors $WE(TF, IDS(tw_i)) =< TF_{ids_{i,1}}, TF_{ids_{i,2}}, \cdots, TF_{ids_{i,m}} >$.
3) For each channel, a sequence of word vectors $DWE(TF, IDS(tw_i)) =< TF_{ids_{i,2-k}}, TF_{ids_{i,3-k}}, \cdots, TF_{ids_{i,m,m-k+1}} >$ is extracted through the embedding layers, where if $ids_{i,j}$ and $j \leq 0$, $TF_{ids_{i,j}}$ is the zero-padding vector. Moreover, if $j > l$, where $l$ is the length of words in $tw_i$, $TF_{ids_{i,j}}$ is also the zero-padding vector.
4) Each time delay sequence of word vectors is referred to as a two-dimensional array. A three-dimensional array is generated by stacking the $k$-extracted two-dimensional arrays.

Fig. 5 illustrates a multi-channel distributed representation. In this example, the number of channels is 3; therefore, four sequences of word vectors are extracted and a $7 \times 4 \times 3$ array is created. Through this process, a word on which the focus is placed can be located in a word vector that appears before it and includes its distributed representation. Hence, this word is expressed with more detailed information.

Fig. 6 shows the proposed model, the input of which is an integer value array numbered for each word in a word string. The dimension of the distributed representation in the embedding layer is set to an integer value $d$. Next, this model is set in a layer that converts the output into multi-channels from a matrix of the distributed representation. This layer’s outputs are passed to the convolution layer and max-pooling layer. Finally, this model’s output is computed through the fully connected layer and activation function, softmax. Each layer is explained as follows. First, in the output conversion layer, the output is stacked to obtain the three dimension. In the convolution layer, the filter size has height $h$, width $d$, and depth $k$. The filter’s width was taken equal to the dimension of the distributed representation to extract the full information of a word.

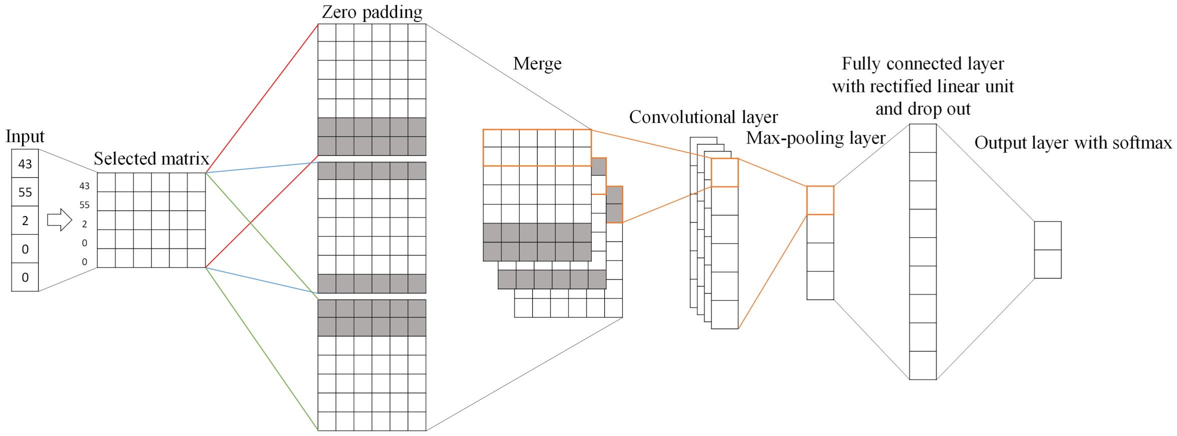


Fig. 6. Proposed model

## VI. Experiments

To evaluate the proposed model, we conducted the following experiments.

TABLE I
Evaluation models : Precision, Recall, F-measure for "relevant" class, and Accuracy in the "rain" dataset

| | *Precision* | *Recall* | *F − measure* | *Accuracy* |
|---|---|---|---|---|
| Naive Bayes | 0.706 | 0.744 | 0.721 | 0.675 |
| CNN (w/o multi-channel) | 0.832 | 0.856 | 0.842 | 0.816 |
| CNN (w/ multi-channel) | 0.832 | 0.868 | 0.849 | 0.822 |
| CNN (Merge-3, w/o multi-channel) | 0.820 | 0.876 | 0.845 | 0.816 |
| CNN (Merge-3, w/ multi-channel) | 0.815 | 0.873 | 0.841 | 0.811 |
| LSTM | 0.799 | 0.880 | 0.834 | 0.800 |

TABLE II
Evaluation models : Precision, Recall, F-measure for "relevant" class, and Accuracy in the "snow" dataset

| | *Precision* | *Recall* | *F − measure* | *Accuracy* |
|---|---|---|---|---|
| Naive Bayes | 0.708 | 0.699 | 0.702 | 0.604 |
| CNN (w/o multi-channel) | 0.794 | 0.893 | 0.840 | 0.775 |
| CNN (w/ multi-channel) | 0.788 | 0.904 | 0.839 | 0.774 |
| CNN (Merge-3, w/o multi-channel) | 0.797 | 0.883 | 0.836 | 0.774 |
| CNN (Merge-3, w/ multi-channel) | 0.798 | 0.880 | 0.838 | 0.826 |
| LSTM | 0.780 | 0.882 | 0.826 | 0.756 |

### A. *Experimental setups*

In the experiments, we compared the tweet classifiers with deep-learning-based classifier using the two types of datasets. Each dataset was constructed using tweets that includes the weather keywords, "rain" and "snow." In this study, the tweet classifier classifies tweets into the two classes, "relevant" and "irrelevant." The "relevant" class' tweets were related to a topic, whereas the "irrelevant" class' tweets were not related to a topic. The "rain" dataset included 1458 tweets belonging to the "relevant" class and 1097 tweets belonging to the "irrelevant" class. In addition, the "snow" dataset included 1648 tweets belonging to "relevant" class and 852 tweets belonging to "irrelevant" class. To extract the features from a tweet, the text in the tweet was separated into words in the Japanese language. Then, we used the MeCab library, which can split text at word level. The following models were evaluated in our experiments using the datasets described above.

- Naive Bayes
  - This model was used for tweet classification in our previous study. Naive Bayes is based on Bayes' theorem. It is a simple classifier and was used as the baseline in our experiments.
- CNN without multi-channel distributed representation
  - This model is similar to Yoon-Kim's model. The height of the filter in the convolutional layer is 3.
- CNN with multi-channel distributed representation
  - This model is the proposed model. This model converts the embedding layer's output to the multi-channel distributed representation using time delay. In addition, the channel number can be set to any value.
- CNN (Merge-3) without multi-channel distributed representation
  - This model includes three simultaneous convolutional layers. The height of the filter is set to 3, 4 or 5. The outputs of the convolution layers are merged.
- CNN (Merge-3) with multi-channel distributed representation
  - This model includes three simultaneous convolutional layers. In addition, this model converts the embedding layer's output to the multi-channel distributed representation using time delay. The height

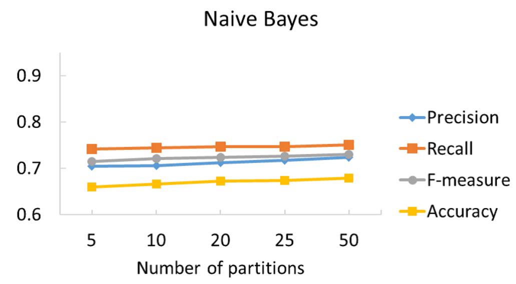


Fig. 7. Evaluation of each number of cross validations (Naive Bayes), when we use “rain” dataset

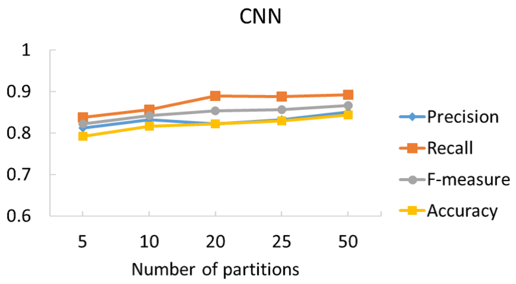


Fig. 8. Evaluation of each number of cross validations (CNN), when we use “rain” dataset

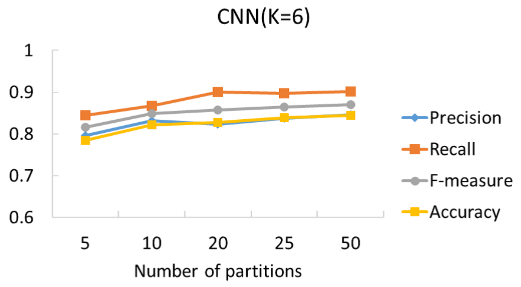


Fig. 9. Evaluation of each number of cross validations (CNN, $k$=6), when we use “rain” dataset

  of the filter is set to 3, 4 or 5. The outputs of the convolution layers are merged.

- Long short-term memory(LSTM) [24]
  - LSTM is based on a recurrent neural networks (RNNs). This model can handle times series data for classification, prediction, and so on. In these experiments, only the final output was passed to the fully-connected layer.

These deep models were constructed using Keras [25], a well-known deep learning framework. We conducted experiments to confirm the performance of the proposed model.

### *B. Experiment 1*

In this experiment, we compared the performance of our models. The hyper parameters for each model were as follows. The length of the input data was set to 80 words and the dimension of the distributed representation was set to 50. When the length of the words in a tweet was less than 80, the empty space in the word identification array was zero-padding. In the CNN models, the number of the filters in convolutional layer was set to 128, and the number of units in the fully-connected layer was set to 128. The number of units in the output layer was set to 2. ReLU was used as the activation function of the convolution and softmax as the activation function of the output layer. To avoid over-fitting, a dropout layer was inserted immediately before the output layer and the unit drop rate was set to 0.5. Each model was trained using cross entropy and Adadelta as the loss function and the optimizer, respectively. The multi-channel number for the CNN model and the CNN (Merge-3) model with the multi-channel method was set to 3.

In the learning and evaluation phase, after the training dataset was split to 10 equal parts, 9 parts were used as the training dataset and the remaining part as the test dataset. Tables I and II show the comparison result of the tweet classifiers for the “rain” dataset and the “snow” dataset, respectively. The deep learning models are more accurate than the naive Bayes classifier. In the results for the “rain” dataset, the CNN model and CNN (Merge-3) model show a better performance than the LSTM model. Moreover, the proposed model showed a good performance as compared to the CNN model and the CNN (Merge-3) model without multi-channel distributed representation. These results show that the proposed model is effective. For the “snow” dataset, the results of the models with multi-channel distributed representation are not more accurately than that of the CNN model and the CNN (Merge-3) model without multi-channel distributed representation. However, according to the precision, recall, and f-measure values of the CNN model, its performance is improved as compared to the model without multi-channel distributed representation.

### *C. Experiment 2*

The objective of Experiment 2 was to confirm the changes in the model’s performance for each number of cross validations. In this experiment, the number of cross validations is set to 2, 10, 20, 25, and 50. Figs. 7, 8, and 9 show the performance of each model on the “rain” dataset. Moreover, Figs. 10, 11, and 12 show the results for the “snow” dataset. Figs. 7 and 10 show the accuracy of the naive Bayes classifier for each the number of cross validations. Figs. 8 and 9 show the accuracy of the CNN model and the CNN(Merge-3) model with multi-channel distributed representation for the “rain” dataset. Figs. 11 and 12 show the results for the “snow” dataset. These results show that the accuracy is improved by increasing the number of cross validations. That is, the deep learning models are expected to improve the performance by increasing the size of the training dataset.

### *D. Experiment 3*

In this experiments, the CNN model and the CNN (Merge-3) model with multi-channel distributed representation were evaluated through a grid search for hyper parameters. We performed a grid search of the number of filters in the convolutional layer and the number of units in the fully-connected layer. The combination of each parameter is shown

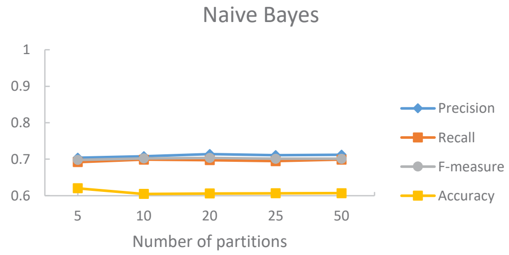


Fig. 10. Evaluation of each number of cross validations (Naive Bayes), when we use "snow" dataset

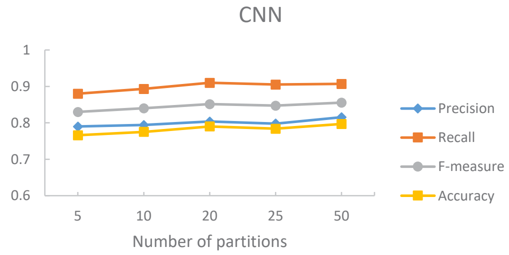


Fig. 11. Evaluation of each number of cross validations (CNN), when we use "snow" dataset

in Table III. Tables IV and V show the results for the "rain" dataset of the CNN model and the CNN(Merge-3) model, set to 3 channel and the number of cross validations to 10. Moreover, Tables VI and VII show the results for the "snow" dataset of the CNN model and the CNN(Merge-3) model, in which the number of channels was set to 3. The best models are the pattern numbers 1, 1, 3 and 4 in each result.

## VII. Conclusion

In this paper, we proposed a new classifier based on deep learning techniques to improve the classification performance of a topic analysis system and presented a detailed evaluation of our model including a comparison of its performance with that of several methods. The main characteristic of the proposed model is the use of the multi-channel distributed representation technique. The proposed model is based on the Kim's model, which utilizes a CNN with distributed representation, a word embedding technique in which words are mapped to vectors in a multi-dimensional space. In Kim's model, text data are converted to a sequence of distributed representations. To enhance the capability of distributed representation, multi-channel distributed representation combines multiple matrices of distributed representation, which are constructed based on time delay. To evaluate the performance of the proposed model, we compared it with that of several deep neural network models. The results showed that the classification performance of the deep learning models was superior to that of the naive Bayes method. Moreover, a CNN with multi-channel distributed representation can classify tweets better than a CNN without multi-channel distributed representation. In our future work, we plan to enhance the representation of input data to improve the proposed model furthers.

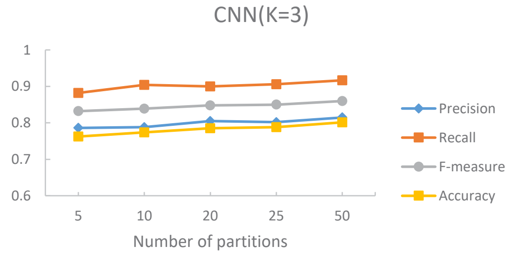


Fig. 12. Evaluation of each number of cross validations (CNN, $k$=6), when we use "snow" dataset

TABLE III
Evaluations of the model parameters pattern 9 ways

| Pattern $n$ | $filters\ in\ CNN\ layer$ | $units\ in\ full\ connected\ layer$ |
|---|---|---|
| Pattern1 | 32 | 32 |
| Pattern2 | 32 | 128 |
| Pattern3 | 32 | 1024 |
| Pattern4 | 128 | 32 |
| Pattern5 | 128 | 128 |
| Pattern6 | 128 | 1024 |
| Pattern7 | 1024 | 32 |
| Pattern8 | 1024 | 128 |
| Pattern9 | 1024 | 1024 |

## Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP18K11320, a Hiroshima City University Grant for Special Academic Research and a grant of the Satake Technical Foundation.

## References

[1] A. Kavanaugh, E. A. Fox, S. Sheetz, S. Yang, L. T. Li, T. Whalen, D. Shoemaker, P. Natsev, and L. Xie, "Social media use by government: From the routine to the critical," in *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, ser. dg.o '11, 2011, pp. 121–130.

[2] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on twitter," in *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, ser. HICSS '10, 2010, pp. 1–10.

[3] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10, 2010, pp. 591–600.

[4] S. Asur and B. A. Huberman, "Predicting the Future with Social Media," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, ser. WI-IAT '10, 2010, pp. 492–499.

[5] A. Java, X. Song, T. Finin, and B. Tseng, "Why We Twitter: Understanding Microblogging Usage and Communities," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, ser. WebKDD/SNA-KDD '07, 2007, pp. 56–65.

[6] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter Power: Tweets As Electronic Word of Mouth," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 11, pp. 2169–2188, Nov. 2009.

[7] T. Sakai and K. Tamura, "Real-time analysis application for identifying bursty local areas related to emergency topics," *SpringerPlus*, vol. 4, no. 1, p. 162, Apr 2015. [Online]. Available: https://doi.org/10.1186/s40064-015-0817-x

[8] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds. ACL, 2014, pp. 1746–1751. [Online]. Available: http://aclweb.org/anthology/D/D14/D14-1181.pdf

TABLE IV
EVALUATION CNN MODELS FOR GRID SEARCHING : PRECISION, RECALL, F-MEASURE FOR "RELEVANT" CLASS, AND ACCURACY IN THE "RAIN" DATASET

| $Pattern$ | $Precision$ | $Recall$ | $F-measure$ | $Accuracy$ |
|---|---|---|---|---|
| 1 | 0.825 | 0.882 | 0.852 | 0.824 |
| 2 | 0.833 | 0.874 | 0.850 | 0.823 |
| 3 | 0.827 | 0.856 | 0.841 | 0.813 |
| 4 | 0.842 | 0.838 | 0.839 | 0.816 |
| 5 | 0.820 | 0.883 | 0.848 | 0.818 |
| 6 | 0.832 | 0.855 | 0.842 | 0.816 |
| 7 | 0.817 | 0.872 | 0.843 | 0.813 |
| 8 | 0.824 | 0.868 | 0.844 | 0.816 |
| 9 | 0.810 | 0.888 | 0.846 | 0.813 |

TABLE V
EVALUATION CNN(MERGE-3) MODELS FOR GRID SEARCHING : PRECISION, RECALL, F-MEASURE FOR "RELEVANT" CLASS, AND ACCURACY IN THE "RAIN" DATASET

| $Pattern$ | $Precision$ | $Recall$ | $F-measure$ | $Accuracy$ |
|---|---|---|---|---|
| 1 | 0.828 | 0.875 | 0.850 | 0.821 |
| 2 | 0.824 | 0.864 | 0.841 | 0.813 |
| 3 | 0.822 | 0.864 | 0.841 | 0.813 |
| 4 | 0.829 | 0.861 | 0.842 | 0.818 |
| 5 | 0.815 | 0.873 | 0.841 | 0.810 |
| 6 | 0.823 | 0.866 | 0.842 | 0.812 |
| 7 | 0.817 | 0.874 | 0.843 | 0.812 |
| 8 | 0.832 | 0.856 | 0.843 | 0.817 |
| 9 | 0.822 | 0.868 | 0.843 | 0.817 |

TABLE VI
EVALUATION CNN MODELS FOR GRID SEARCHING : PRECISION, RECALL, F-MEASURE FOR "RELEVANT" CLASS, AND ACCURACY IN THE "SNOW" DATASET

| $Pattern$ | $Precision$ | $Recall$ | $F-measure$ | $Accuracy$ |
|---|---|---|---|---|
| 1 | 0.789 | 0.898 | 0.840 | 0.773 |
| 2 | 0.790 | 0.894 | 0.836 | 0.770 |
| 3 | 0.788 | 0.904 | 0.841 | 0.775 |
| 4 | 0.792 | 0.895 | 0.838 | 0.773 |
| 5 | 0.788 | 0.904 | 0.839 | 0.774 |
| 6 | 0.789 | 0.893 | 0.837 | 0.770 |
| 7 | 0.776 | 0.930 | 0.845 | 0.773 |
| 8 | 0.786 | 0.903 | 0.841 | 0.772 |
| 9 | 0.786 | 0.897 | 0.835 | 0.768 |

TABLE VII
EVALUATION CNN(MERGE-3) MODELS FOR GRID SEARCHING : PRECISION, RECALL, F-MEASURE FOR "RELEVANT" CLASS, AND ACCURACY IN THE "SNOW" DATASET

| $Pattern$ | $Precision$ | $Recall$ | $F-measure$ | $Accuracy$ |
|---|---|---|---|---|
| 1 | 0.802 | 0.878 | 0.837 | 0.776 |
| 2 | 0.795 | 0.880 | 0.834 | 0.769 |
| 3 | 0.793 | 0.899 | 0.843 | 0.778 |
| 4 | 0.790 | 0.909 | 0.846 | 0.779 |
| 5 | 0.798 | 0.880 | 0.838 | 0.772 |
| 6 | 0.799 | 0.896 | 0.841 | 0.779 |
| 7 | 0.793 | 0.891 | 0.837 | 0.773 |
| 8 | 0.802 | 0.884 | 0.840 | 0.779 |
| 9 | 0.782 | 0.902 | 0.837 | 0.768 |

[10] S. Hashida, K. Tamura, and T. Sakai, "Multi-channel distributed representation for classifying tweets by using convolutional neural networks," in *Proceedings of The International MultiConference of Engineers and Computer Scientists 2018, 14-16 March, 2018, Hong Kong*, 2018, pp. 278–283.

[11] N. F. F. D. Silva, L. F. S. Coletta, and E. R. Hruschka, "A survey and comparative study of tweet sentiment analysis via semi-supervised learning," *ACM Comput. Surv.*, vol. 49, no. 1, pp. 15:1–15:26, Jun. 2016.

[12] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING '10, 2010, pp. 241–249.

[13] E. Aramaki, S. Maskawa, and M. Morita, "Twitter catches the flu: Detecting influenza epidemics using twitter," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11, 2011, pp. 1568–1576.

[14] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '15, 2015, pp. 959–962.

[15] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, ser. SETN '18. New York, NY, USA: ACM, 2018, pp. 35:1–35:6. [Online]. Available: http://doi.acm.org/10.1145/3200947.3208069

[16] A. Rios and R. Kavuluru, "Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles," in *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, ser. BCB '15. New York, NY, USA: ACM, 2015, pp. 258–267. [Online]. Available: http://doi.acm.org/10.1145/2808719.2808746

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944966

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013. [Online]. Available: http://arxiv.org/abs/1310.4546

[19] L. Xu, C. Jiang, Y. Ren, and H.-H. Chen, "Microblog dimensionality reduction - a deep learning approach." *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1779–1789, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/tkde/tkde28.html

[20] K. Shuang, X. Ren, J. Chen, X. Shan, and P. Xu, "Combining word order and cnn-lstm for sentence sentiment classification," in *Proceedings of the 2017 International Conference on Software and e-Business*, ser. ICSEB 2017. New York, NY, USA: ACM, 2017, pp. 17–21. [Online]. Available: http://doi.acm.org/10.1145/3178212.3178230

[21] T. Sakai, K. Tamura, and H. Kitakami, "Emergency situation awareness during natural disasters using density-based adaptive spatiotemporal clustering," in *Database Systems for Advanced Applications, DASFAA 2015 International Workshops, SeCoP, BDMS, and Posters, Hanoi, Vietnam, April 20-23, 2015*, vol. 9052, 2015, pp. 155–169.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[23] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13, 2013, pp. 3111–3119.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[25] F. Chollet *et al.*, "Keras," https://keras.io, 2015.