Compare Classification Performance of Real Anger Emotion by using Shallow Neural Network and Deep Learning

Hao Wen
Research School of Computer Science
Australian National University
Acton ACT 2601 Australia
U5883475@anu.edu.au

Abstract. Detecting the actual perception of the emotion expressed by human face is important and it can effectively improve the diagnostic of mental health. The diameter change pattern of pupillary response can be used to classify whether a person is watching a real or posed anger. Shallow neural network with Casper algorithm and deep learning including CNN and LSTM are used to construct the classification model. Three main results were found: 1. deep learning has worse prediction performance than shallow neural network and research paper[2] in this case. 2. Pre-padding can improve LSTM model prediction performance. 3. CNN has better prediction performance than LSTM for this case.

Keywords: Shallow Neural Network, Deep Learning, Casper algorithm, CNNs, LSTMs, padding

1. Introduction

Over the past few years, successfully classifying acted and genuine emotion is becoming a popular topic and there have been a number of researches made to classifying posed and real emotion. Classifying posed and real smiles has been proved successfully using Pupillary Response(PR) features and classification accuracy was 93.7% with trained machine classifiers[1]. Another study shows the similar result when detecting real or posed anger by using same Pupillary Response(PR) features and machine classifiers[2]. The result from these two studies suggest pupillary response can be used to predict real and posed emotion and reach a higher accuracy than human verbal response.

Shallow Neural Network has been a powerful tool for many classification problems. Based on the shallow neural network, a new network algorithm called Casper[3] was introduced and is shown to produce more compact networks and better performance on some classification problems. Casper is a constructive learning algorithm which builds cascade networks[5] and employs Progressive RPROP[6] to train the whole network. Compared to traditional shallow neural network, there are two main advantages of Casper algorithm. Firstly, neurons are added one at a time and are connected to all previous hidden and input neurons. The second advantage is Casper doesn't freeze weights and uses modified version of Resilient Back Propagation(RPROP)[6] algorithm to train the whole network. RPROP[6] only uses the sign of the gradient and assume the different weights need different step sizes for update, so it considerably accelerates backpropagation learning and can determine the appropriate step size by itself. While RPROP[6] is fast on converging, it suffers from local minima problem. Another study[4] shows SARPROP algorithm, which is RPROP algorithm with Simulated Annealing term, can address local minima problem and increase the rate of convergence.

Shallow neural network used to describe neural network that has only one hidden layer while deep learning has more than one hidden layer. Deep learning is able to understand the world through a hierarchy of concept and extract better features to represent data through multiple levels of abstraction, so usually deep learning with right architectures achieve better results than shallow neural networks. Deep learning model has many different types including Convolutional Neural Networks(CNNs)[13], Recurrent Neural Networks(RNNs)[10], Lone Short-Term Memory Networks(LSTMs)[9] and so on. Deep learning model has been successfully applied to various area and obtain a good result. CNNs[13] is shown to have the capability to extract a hierarchical feature representation that facilitates categorization[7] and speech emotion recognition[8]. RNNs[10] include cyclic connections that make model is good at modelling sequence data and LSTMs[9] has long-range dependencies that is more accurate than conventional RNNs[10]. LSTMs have shown great performance on various sequence prediction in learning context-free and context-sensitive language[11]. Because the input data in this paper is sequence, so CNNs[13] and LSTMs[9] was used build classification model. In addition, due to the implementation requirement in PyTorch, LSTMs[9] and CNNs[13] need to take input with same length and dimension. Padding is required to pre-process input sequence data. A study[12] shows pre-padding method has better performance than post-padding on LSTM but it doesn't matter to CNN.

Since the main task of this paper is to distinguish the real and posed anger directly from the change of pupil size, the continuous pupil diameter is input and output is classification whether the participate is watching real or posed anger. In addition, this study aims to (1) investigate if deep learning has better prediction performance than shallow neural network. (2) examine the effect of pre-padding and post-padding on LSTM and CNN. (3) compare the prediction performance of LSTM and CNN.

2. Datasets Description and Pre-Processing

Datasets used in this paper called Anger_v2 that comes from a designed experiment[2]. There are 20 participants and each participant watch 20 videos including 10 genuine(True) anger scene and 10 acted (False) anger scene. Left and Right eye pupil size(diameter) were tracked by a remote Eye Tribe eye gaze tracker. Each dataset contains 400 samples where each sample is a sequence that includes the continuous left or right pupil size over the whole video. Because video has different length, each sequence has different length for different video. Sequence label was defined by the video label where "T" means True and is coded as 1, "F" means False and is coded as 0.

Data pre-processing include dropping columns with missing value and interpolating zero value. In both left and right eye dataset, there are 10 samples don't have any response value and these 10 samples are dropped. Another issue in this dataset is zero value of pupil size due to occasional eye blink. Among left and right eye dataset, 176 sequences include zero value and for each sequence, the biggest proportion of zero is 57.3%. These zero values were imputed by linear interpolation between the nearby nonzero values. Figure 1 shows the result of before and after imputation on a single sequence from left eye dataset.

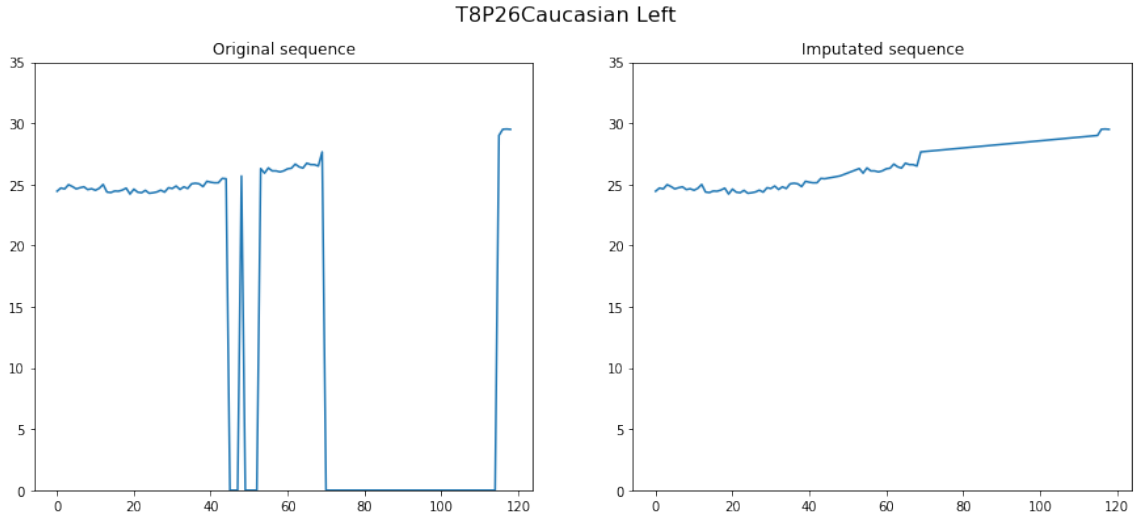


Figure 1. Before and after imputation

3. Method

3.1 Shallow Neural Network

3.1.1 Casper algorithm

The Casper algorithm construct the network in a similar way to Cascor[5]: starts with a single hidden neuron as initial network and successively adds single hidden neuron from candidate pool to the previous network each time. The new hidden neuron should connect all input neurons and previous hidden neurons. And then use backpropagation algorithm update different weights with different learning rates and train the whole network. There are three different learning rates for Casper algorithm as shown in Figure 2. Learning rate L1 is for region 1 where the weights connect all inputs and hidden neurons to new hidden neuron. Learning rate L2 is for region 2 where the weights connected from new hidden neuron to output neurons. Learning rate L3 is for region 3 where include all the old weights from previous network. Usually the relative value is L1>>L2>L3. The reason is high value of L1 allows the new hidden neuron to learn quickly the remaining network error. At the same time, L2 and L3 allows the other neurons to reduce the network loss as well but with little interference.

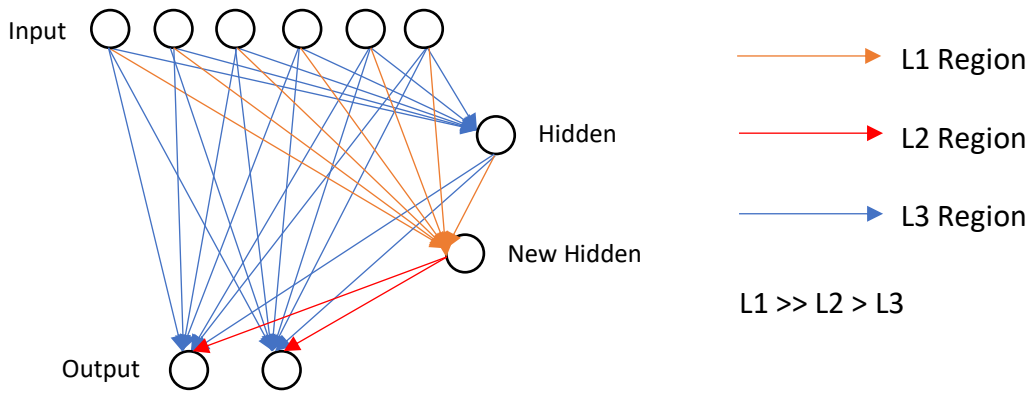


Figure 2. Casper algorithm

3.1.2 RPROP and SARPROP algorithm

RPROP[6] is a gradient descent algorithm only uses the sign of the gradient. Also, it assumes the different weights need different step size for updates, which vary throughout the process. The basic idea is if the error gradient for a given weight had the same sign in two consecutive epochs, we increase its step size because the optimal value may be far away. On the other hand, we decrease the step size if the sign switched. Finally update weights with step size. RPROP algorithm can eliminate the noisy effect of the size of gradient and avoid getting stuck with extreme weights because of shallow slope in the activation function.

SARPROP[4] algorithm is based on RPROP[6] algorithm and makes use of weight decay (Simulated Annealing term) as a mean to increase the rate of convergence for some problem and avoid local minima. There are two main enhancements. Firstly, a noise factor is introduced. Noise is added to a weight when both the error gradient changes sign in successive epoch and the magnitude of the update value is less than a value proportional to the current loss. This will allow the weight to jump out of local minima. Secondly, Weight decay term was added to the error function.

3.2 Deep Learning

3.2.1 Padding

Pre-padding

All the sequences are padded with zero value in the beginning of the sequence according to the longest sequence length.

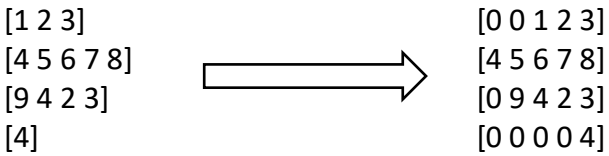


Figure 3. Pre-padding

Post-padding

All the sequences are padded with zero value in the ending of the sequence according to the longest sequence length.

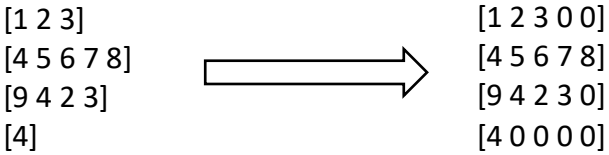


Figure 4. Post-padding

3.2.2 Convolutional neural network

A convolutional neural network[13] consists of an input layer, multiple hidden layers and an output layer. The hidden layers typically consist of a series of convolutional layers and pooling layers.

Convolutional layers

Convolutional layers are the core part of a CNN[13]. Convolutions is an operation which describe the mixing of two functions. In convolutional layer, filters are used to detect feature of input by convolving with input data. The current part of the input that is being convolved with the filter is called local receptive field. Each filter slide over all receptive fields and convolve with input to produce a series of neurons in a feature map. Also, stride length control how far local receptive field slides. The output of convolution layer is feature maps. The number of feature maps is the number of filters and each feature map represents a feature. The feature map usually is the input to the next layer such as pooling layer.

Pooling layer

Pooling layer is used to produce a summary statistic of the feature map from the previous convolution layer. It can down sample feature map into a condensed version and assist in controlling overfitting, so the learned feature representation is invariant to small translation of input. There are few different ways to pooling such as max-pooling, L2-pooling and average-pooling. Figure 5 shows the CNN model used in this paper.

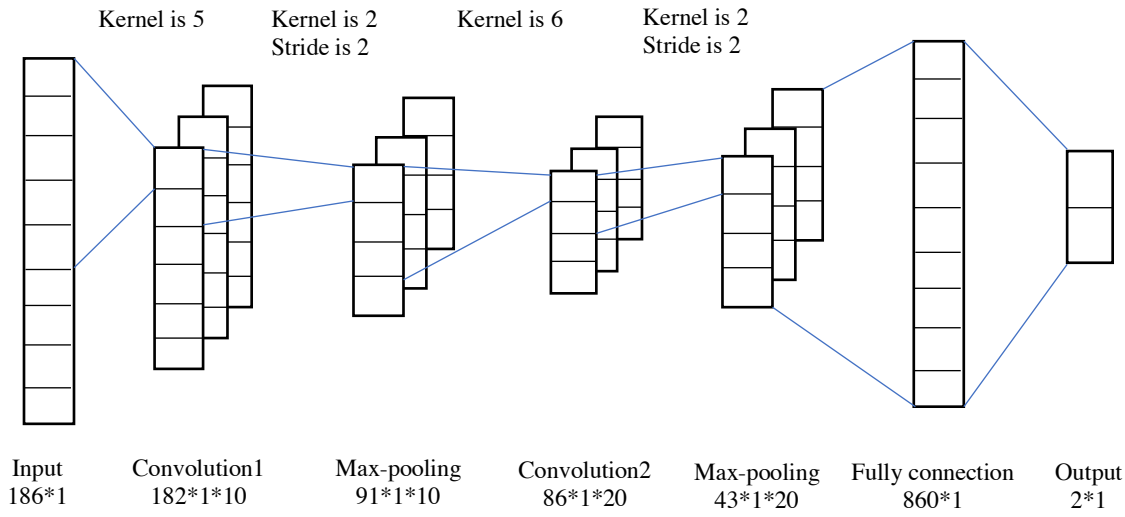


Figure 5. CNN Architecture

3.2.3 Long short-term memory (LSTM)

LSTMs[9] are a modification of recurrent neural networks (RNNs)[10]. RNNs consider their previous output as an input along with the next input, this allows model keep track of previous output and making model good to work with sequences(Figure 6). Compared to standard feed forward Neural Network, hidden state of RNN holds the memory of the network and the representation of the current input. LSTM is a special case of RNN where the hidden layer units are replaced by memory blocks. Each memory block contains one or more memory cells along with input, output and forget gates which control flow of information into and out of the memory cell.

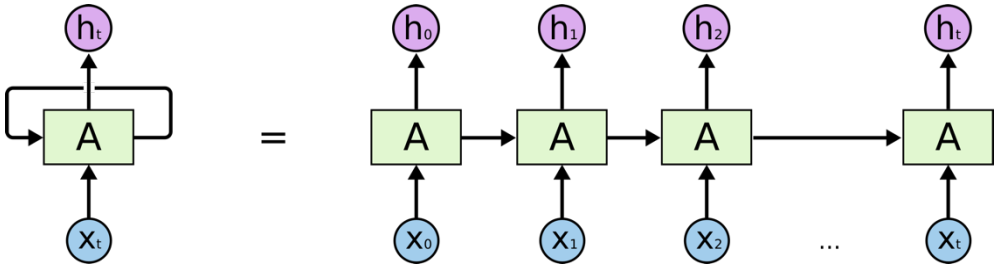
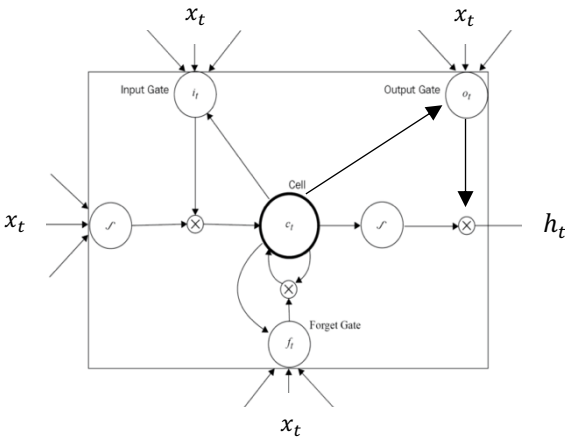


Figure 6. RNN
https://colah.github.io/posts/2015-08-Understanding-LSTMs/



$$i_t = sigmoid\left(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}\right) \quad (1)$$
$$f_t = sigmoid(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf}) \quad (2)$$
$$g_t = tanh(W_{ig}x_t + b_{ig} + W_{hc}h_{(t-1)} + b_{hg}) \quad (3)$$
$$o_t = sigmoid(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}) \quad (4)$$
$$c_t = f_t * c_{(t-1)} + i_t * g_t \quad (5)$$
$$h_t = o_t * \tanh(c_t) \quad (6)$$

Figure 7. LSTM

The basic idea of LSTM is that the memory cell stores information until it's relevant. Figure 7 shows an example of a memory block in LSTM. The input gate i_t controls what part of new input goes into the current hidden unit based on the previous content of the memory cell and hidden unit and the current input. The output gate o_t controls what comes out of the memory cell into the current hidden unit. The forget gate f_t controls when the contents of the memory cell are forgotten. The input modulation gate g_t modulates the information that goes into the memory cell allowing for faster convergence.

4. Model Design

4.1 Implementing shallow neural network with Casper algorithm

The main idea of Casper algorithm is adding one hidden unit each time to network until it reaches a predefined number of hidden unit, and record the classification accuracy on testing data for each adding hidden unit. For example, the predefined number of hidden units is 10. Add first hidden neuron to Casper network, record the accuracy on testing data, and then adding the second hidden neuron and record the accuracy, keep adding until it adds to 10 hidden neurons. After that, find out at which hidden neuron the network achieves the highest accuracy. This hidden unit and accuracy will be regarded as the best hidden neuron and best accuracy. Repeat the whole process ten times and take average on best hidden neuron and best accuracy.

4.2 Selecting appropriate number of hidden neurons and learning rate

The pre-processed left eye data with post padding was divided into training data and validation data. Four videos were randomly selected as validation label, which includes the third true vide(T3), the third false vide(F3), the eighth true vide(T8), the eighth false vide(F8). The sample sequences include validation label are validation data and other sample sequences are training data. There are seven candidate learning rate including 0.0001, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3 and five candidate number of hidden neurons including 1, 5, 10, 20, 40. The other settings of LSTM model are number of layer is 1, number of classes is 2, input size is 1, number of epoch is 300, loss function is CrossEntropyLoss and optimizer is Rprop. Train the LSTM model on training data and select the best hyperparameter with the best accuracy on validation data. The result shows the best learning rate is 0.003 and best number of hidden neurons is 20.

4.3 Experiment details

After determining the hyperparameters, 5 cross validation was used to measure the model performance. The dataset is the mean of left and right eye dataset with pre-padding and post-padding, the video label is used as the index to separate data. For example, the sample sequence including T1, T2, F1 and F2 are testing data and other sample sequences are training data. Similarly, T3, T4, F3 and F4 are testing label and so on. CNN model was defined in 3.2.2 Figure 5. Apply the LSTM model and CNN model on training data with epoch 100 and then calculate prediction accuracy on testing data. Model performance is measured by prediction accuracy on testing data. The higher prediction accuracy is, the better the network is.

5 Result and Discussion

The table 1 shows the prediction accuracy on testing data using deep learning. CNN has mean prediction accuracy 80.57% with post-padding and 79.3% with pre-padding while LSTM only has 49.46% and 60.21% respectively. This result shows

CNN has better prediction performance than LSTM. Also, for LSTM model, pre-padding method has mean prediction accuracy 60.21% while post-padding only has 49.46%, this result shows pre-padding is better than post-padding for LSTM model. However, there is not much difference for CNN model between pre-padding and post-padding. This result is same in other research[12]. In addition, this table shows a failure result of the LSTM model. The prediction accuracy is between 50% to 60% which is slightly higher than a random guess 50% and similar with verbal response 60%[2]. Table 2 show the result of shallow neural network and results from other research[2]. Shallow neural network using Casper algorithm and machine classifier[2] have mean accuracy 94% and 95% respectively, which is higher than deep learning.

To gain the deep understanding of the reason for the failure of LSTM, the prediction output was collected and they show only one label (either 1 or 0) was predicted for all testing sample sequence. This demonstrates LSTM suffer a serious local minima issue. Weight decay and momentum are used in the experiment to avoid local minima problem, but this doesn't help much. Another reason for failure is LSTM is not feasible for this task. LSTM try to memory information from previous output in each timestep of a sequence but actually in this task, only few patterns existed in sequence is useful for classification, not the whole sequence. So CNN can detect this pattern in the sequence and has better performance than LSTM.

Table 1

Testing data	LSTM (Post-padding)	LSTM (Pre-padding)	CNN (Post-padding)	CNN (Pre-padding)
T1, T2, F1, F1	0.5	0.7571	1	1
T3, T4, F3, F4	0.4933	0.5733	0.7467	0.7467
T5, T6, F5, F6	0.4865	0.5135	0.7432	0.7568
T7, T8, F7, F8	0.4933	0.6667	1	1
T9, T10, F9, F10	0.5	0.5	0.5385	0.4615
Mean accuracy	49.46%	60.21%	80.57%	79.3%

Table 2

	Shallow neural network (Casper network)	Machine classifiers [2]	Verbal response [2]	Random guess
Mean accuracy	94%	95%	60%	50%

6 Conclusion and Future Work

In summary, this paper showed that deep learning doesn't have better classification performance than shallow neural network and the result is even worse on this task. Also compared to post-padding, pre-padding can improve the performance of LSTM model. However, pre-padding and post-padding doesn't matter much to CNN. Finally, CNN has better performance than LSTM on this task.

There are some limitations in this paper. Sequence is too long and contains too much unimportant information which makes LSTM difficult to learn the important feature for classification. The LSTM would have better result if the sequence can be reduced and mainly contain pupil value when seeing real anger or posed anger scene. Another limitation is CNN is very sensitive to the test dataset. On some testing dataset, the accuracy is very high that is 100% while on other testing dataset the accuracy is just 53.85%. This shows CNN is able to detect features but the sample size is too small that makes result unstable. If the sample is big, the CNN would have better prediction performance. In the future, more advanced technique such as attention mechanism should try on LSTM to deal with local minima issue. Transfer learning can be applied on CNN to deal with the small sample size issue. Casper algorithm can be combined with fully connected layer in CNN.

References

1. Md Zakir Hossain, Tom Gedeon. Classifying posed and real smiles from observers' peripheral physiology. Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare. May 2017. Pages 460–463 (2017).
2. Lu Chen, Tom Gedeon, Md Zakir Hossain, Sabrina Caldwell Are you really angry? Detecting emotion veracity as a proposed tool for interaction. Proceedings of the 29th Australian Conference on Computer-Human Interaction November 2017 Pages 412–416 (2017).

3. Treadgold N.K., Gedeon T.D. A cascade network algorithm employing Progressive RPROP. In: Mira J., Moreno-Díaz R., Cabestany J. (eds) Biological and Artificial Computation: From Neuroscience to Technology. IWANN (1997).
4. Treadgold, N. & Gedeon, Tom. The SARPROP Algorithm: A Simulated Annealing Enhancement To Resilient Back Propagation (1997).
5. Fahlman, S.E., and Lebiere, C. The cascade-correlation learning architecture. In Advances in Neural Information Processing II, Touretzky, Ed. San Mateo, CA: Morgan Kauffman, 1990, pp. 524-532 (1990).
6. Riedmiller, M. and Braun, H. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Ruspini, H., (Ed.) Proc. of the ICNN 93, San Francisco, pp. 586-591(1993).
7. D. Yu, M. L. Seltzer, J. Li, J. T. Huang, and S. Frank, Feature learning in deep neural networks - studies on speech recognition tasks, in ICLR, Scottsdale, Arizona, USA, May (2013).
8. Zhengwei Huang y , Ming Dong z , Qirong Mao y , Yongzhao Zhan y, Speech Emotion Recognition Using CNN, MM '14: Proceedings of the 22nd ACM international conference on MultimediaNovember Pages 801–804 (2014).
9. S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. (1997).
10. Y. Bengio, P. Simard, and P. Frasconi, Learning long-term de- pendencies with gradient descent is difficult, Neural Networks, IEEE Transactions on, vol. 5, no. 2, pp. 157–166, (1994).
11. F. A. Gers and J. Schmidhuber, LSTM recurrent networks learn simple context free and context sensitive languages, IEEE Trans- actions on Neural Networks, vol. 12, no. 6, pp. 1333–1340, (2001).
12. Dwarampudi, Mahidhar and N. V. Subba Reddy. Effects of padding on LSTMs and CNNs. ArXiv abs/1903.07288 (2019).
13. LeCun, Y. and Bengio, Y., Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), (1995).