KINN: Incorporating Expert Knowledge in Neural Networks

Muhammad Ali Chattha[123], Shoaib Ahmed Siddiqui[12], Muhammad Imran Malik[34], Ludger van Elst[1], Andreas Dengel[12], Sheraz Ahmed[1]

[1]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany.
[2]TU Kaiserslautern, Kaiserslautern, Germany.
[3]School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST), Islamabad, Pakistan.
[4]Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad, Pakistan.

Abstract

The ability of Artificial Neural Networks (ANNs) to learn accurate patterns from large amount of data has spurred interest of many researchers and industrialists alike. The promise of ANNs to automatically discover and extract useful features/patterns from data without dwelling on domain expertise although seems highly promising but comes at the cost of high reliance on large amount of accurately labeled data, which is often hard to acquire and formulate especially in time-series domains like anomaly detection, natural disaster management, predictive maintenance and healthcare. As these networks completely rely on data and ignore a very important modality i.e. expert, they are unable to harvest any benefit from the expert knowledge, which in many cases is very useful. In this paper, we try to bridge the gap between these data driven and expert knowledge based systems by introducing a novel framework for incorporating expert knowledge into the network (KINN). Integrating expert knowledge into the network has three key advantages: (a) Reduction in the amount of data needed to train the model, (b) provision of a lower bound on the performance of the resulting classifier by obtaining best of both worlds, and (c) improved convergence of model parameters (model converges in lesser number of epochs). Although experts are extremely good in solving different tasks, there are some trends and patterns, which are usually hidden only in the data. Therefore, KINN employs a novel residual knowledge incorporation scheme, which can automatically determine the quality of the predictions made by the expert and rectify it accordingly by learning the trends/patterns from data. Specifically, the method tries to use information contained in one modality to complement information missed by the other. We evaluated KINN on a real world traffic flow prediction problem. KINN significantly superseded performance of both the expert and as well as the base network (LSTM in this case) when evaluated in isolation, highlighting its superiority for the task.

Deep Neural Networks (DNNs) have revolutionized the domain of artificial intelligence by exhibiting incredible performance in applications ranging from image classification (Krizhevsky, Sutskever, and Hinton 2012), playing board games (Silver et al. 2016), natural language processing (Conneau et al. 2017) to speech recognition (Hinton et al. 2012). The biggest highlight of which was perhaps Google DeepMind's AlphaGo system, beating one of the world's best Go player, Lee Sedol in a 5 series match (Wang et al. 2016). Consequently, the idea of superseding human performance has opened a new era of research and interest in artificial intelligence. However, the success of DNNs overshadows its limitations. Arguably the most severe limitation is its high reliance on large amount of accurately labeled data which in many applications is not available (Sun et al. 2017). This is specifically true in domains like anomaly detection, natural disaster management and healthcare. Moreover, training a network solely on the basis of data may result in poor performance on examples that are not or less often seen in the data and may also lead to counter intuitive results (Szegedy et al. 2013).

Humans tend to learn from examples specific to the problem, similar to DNNs, as well as from different sources of knowledge and experiences (Lake, Salakhutdinov, and Tenenbaum 2015). This makes it possible for humans to learn just from acquiring knowledge about the problem without even looking at the data pertaining to it. Domain experts are quite proficient in tasks belonging to their area of expertise due to their extensive knowledge and understanding of the problem, which they have acquired overtime through relevant education and experiences. Hence, they rely on their knowledge when dealing with problems. Due to their deep insights, expert predictions even serve as a baseline for measuring the performance of DNNs. Nonetheless, it can not be denied that apart from knowledge, the data also contains some useful information for solving problems. This is particularly cemented by astonishing results achieved by the DNNs that solely rely on data to find and utilize hidden features contained in the data itself (Krizhevsky, Sutskever, and Hinton 2012).

Therefore, a natural step forward is to combine both these separate streams of knowledge i.e. knowledge extracted from the data and the expert's knowledge. As a matter of fact, supplementing DNNs with expert knowledge and predictions in order to improve their performance has been actively researched upon. A way of sharing knowledge among classes in the data has been considered in zero-shot-learning (Rohrbach, Stark, and Schiele 2011), where semantic relatedness among classes is used to find classes related to the

Copyright held by the author(s). In A. Martin, K. Hinkelmann, A. Gerber, D. Lenat, F. van Harmelen, P. Clark (Eds.), Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019). Stanford University, Palo Alto, California, USA, March 25-27, 2019.

known ones. Although such techniques employ knowledge transfer, they are restricted solely to the data domain and the knowledge is extracted and shared from the data itself without any intervention from the expert. Similarly, expert knowledge and opinions are incorporated using distillation technique where expert network produces soft predictions that the DNN tries to emulate or in the form of posterior regularization over DNN predictions (Hinton, Vinyals, and Dean 2015). All of these techniques try to strengthen DNN with expert knowledge. However, cases where the expert model is unreliable or even random have not been considered. Moreover, directly trying to mimic expert network predictions has an implicit assumption regarding the high quality of the predictions made by the expert. We argue that the ideal incorporation of expert network would be the one where strengths of both networks are promoted and weaknesses are suppressed. Hence, we introduce a step in this direction by proposing a novel framework, Knowledge Integrated Neural Network (KINN), which aims to integrates knowledge residing in heterogeneous sources, in the form of predictions, in a residual scheme KINN's design allows it to be flexible. KINN can successfully integrate knowledge in cases where predictions of the expert and DNN align and as well as in scenarios where they are completely disjoint. Finding state-of-the-art DNN or expert model is not the aim here but rather, the aim is to devise a strategy that facilitates integration of expert knowledge with DNNs in a way that the final network achieves best of both worlds.

The residual scheme employed in KINN to incorporate expert knowledge inside the network has three key advantages: (a) Significant reduction in the amount of data needed to train the model, since the network has to learn a residual function instead of learning the complete input to output space projection, (b) a lower bound on the performance of KINN based on the performance of the two subsequent classifiers achieving the best of both worlds, and (c) improvements in convergence of the model parameters as learning a residual mapping makes the optimization problem significantly easier to tackle. Moreover, since the DNN itself is data driven, this makes KINN robust enough to deal with situations where the predictions made by the expert model are not reliable or even useless.

The rest of the paper is structured as follows: We first provide a brief overview of the work done in the direction of expert knowledge incorporation in the past. We then explain the proposed framework, KINN, in detail. After that, we present the evaluation results regarding the different experiments performed in order to prove the efficacy of KINN for the task of expert knowledge incorporation. Finally, we conclude the paper with the conclusion.

Related Work

Integrating domain knowledge and experts opinion into the network is an active area of research and even dates back to the early 90s. Knowledge-based Artificial Neural Networks (KBANN) was proposed by (Towell and Shavlik 1994). KBANN uses knowledge in the form of propositional rule sets which are hierarchically structured. In addition to directly mapping inputs to outputs, the rules also state intermediate conclusions. The network is designed to have a one-to-one correspondence with the elements of the rule set, where neurons and the corresponding weights of their connections are specified by the rules. Apart from these rule based connections and neurons, additional neurons are also added to learn features not specified in the rule set. Similar approach has also been followed by (Tran and Garcez 2018). Although such approaches directly incorporates knowledge into the network, but they also limit the network architecture by forcing it to have strict correspondence with the rule base. As a result, this restricts the use of alternate architectures or employing network that does not directly follow the structure defined by the rule set.

(Hu et al. 2016) integrated expert knowledge using first order logic rules which is transferred to the network parameters through iterative knowledge distillation (Hinton, Vinyals, and Dean 2015). The DNN tries to emulate soft predictions made by the expert network, instilling expert knowledge into the network parameters. Hence, the expert network acts as a teacher to the DNN i.e. the student network. The objective function is taken as a weighted average between imitating the soft predictions made by the teacher network and true hard label predictions. The teacher network is also updated at each iteration step with the goal of finding the best teacher network that fits the rule set while, at the same time, also staying close to the student network. In order to achieve this goal, KL-divergence between the probability distribution of the predictions made by the teacher network and softmax output layer of the student network is used as the objective function to be minimized. This acts as a constraint over model posterior. The proposed framework was evaluated for classification tasks and achieved superior results compared to other state-of-the-art models at that time. However, the framework strongly relies on the expert network for parametric optimization and does not cater for cases where expert knowledge is not comprehensive.

Expert knowledge is incorporated for key phrase extraction by (Gollapalli, Li, and Yang 2017) where they defined label-distribution rules that dictates the probability of a word being a key phrase. For example, the rule enunciates that a noun that appears in the document as well as in the title is 90% likely to be a key phrase and thus acts as posterior regularization providing weak supervision for the classification task. Similarly, KL-divergence between the distribution given by the rule set and the model estimates is used as the objective function to be used for the optimization. Again, as the model utilizes knowledge to strengthen the predictions of the network, it shifts the dependency of the network from the training data to accurate expert knowledge which might just be an educated guess in some cases. Similarly, (Xu et al. 2017) incorporated symbolic knowledge into the network by deriving a semantic loss function that acts as a bridge between the network outputs and the logical constraints. The semantic loss function is based on constraints in the form of propositional logic and the probabilities computed by the network. During training, the semantic loss is added to the normal loss of the network and thus acts as a regularization term. This ensures that symbolic knowledge plays a part in updating the parameters of the network.

(Wu et al. 2016) proposed a Knowledge Enhanced Hybrid Neural Network (KEHNN). KEHNN utilizes knowledge in conjunction with the network to cater for text matching in long texts. Here, knowledge is considered to be the global context such as topics, tags etc. obtained from other algorithms that extracts information from multiple sources and datasets. They employed the twitter LDA model (Zhao et al. 2011) as the prior knowledge which was considered useful in filtering out noise from long texts. A special gate known as the knowledge gate is added to the traditional bi-directional Gated Recurrent Units (GRU) in the model which controls how much information from the expert knowledge flows into the network.

KINN: The Proposed Framework

Problem Formalization

Time-series forecasting is of vital significance due to its high impact, specifically in domains like supply chain (Fildes, Goodwin, and Onkal 2015), demand prediction (Pacchin et al. 2017), and fault prediction (Baptista et al. 2018). In a typical forecasting setting, a sequence of values $\{x_{t-1}, x_{t-2}, ..., x_{t-p}\}$ from the past are used to predict the value of the variable at time-step t, where p is the number of past values leveraged for a particular prediction, which we refer as the window size. Hence, the model is a functional mapping from past observations to the future value. This parametric mapping can be written as:

$$\hat{x}_t = \phi([x_{t-1}, x_{t-2}, ..., x_{t-p}]; \mathcal{W})$$

where $\mathcal{W} = \{W_l, b_l\}_{l=1}^{L}$ encapsulates the parameters of the network and $\phi : \mathbb{R}^p \mapsto \mathbb{R}$ defines the map from the input space to the output space. The optimal parameters of the network $\mathcal{W}^*$ are computed based on the empirical risk computed over the training dataset. Using MSE as the loss function, the optimization problem can be stated as:

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} (x_t - \phi([x_{t-1}, ..., x_{t-p}]; \mathcal{W}))^2 \tag{1}$$

where $\mathcal{X}$ denotes the set of training sequences and $\mathbf{x} \in \mathbb{R}^{p+1}$. Solving this optimization problem comprising of thousands, if not millions of parameters, requires large amount of data in order to successfully constrain the parametric space so that a reliable solution is obtained.

Humans on the other hand, leverage their real-world knowledge along with their past-experiences in order to make predictions about the future. The aim of KINN is to inject this real-world knowledge in the form of expert into the system. However, as mentioned, information from the expert may not be reliable, therefore, KINN proposes a novel residual learning framework for the incorporation of expert knowledge into the system. The residual framework conditions the prediction of the network on the expert's opinion. As a result, the network acts as a correcting entity for the values generated by the expert. This decouples our system from complete reliance on the expert knowledge.

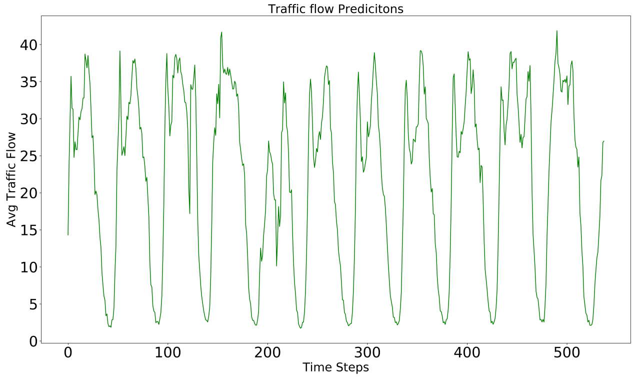


Figure 1: Traffic flow data grouped into 30 minute windows

Dataset

We evaluated KINN on Caltrans Performance Measurement System (PeMS) data. The data contains records of sensor readings that measure the flow of vehicular traffic on California Highways. Since the complete PeMS dataset is enormous in terms of its size comprising of records from multiple highways, we only considered a small fraction of it for our experiments i.e. the traffic flow on Richards Ave, from January 2016 till March 2016[1]. The dataset contains information regarding the number of vehicles passing on the avenue every 30 seconds. PeMS also contains other details regarding the vehicles, however, we only consider the problem of average traffic flow forecasting in this paper. The data is grouped into 30 minute windows. The goal is to predict average number of vehicles per 30 seconds for the next 30 minutes. Fig. 1 provides an overview of the grouped dataset. The data clearly exhibits a seasonal component along with high variance for the peaks.

Baseline Expert and Deep Models

LSTMs have achieved state-of-the-art performance in a range of different domains comprising of sequential data such as language translation (Weiss et al. 2017), and handwriting and speech recognition (Zhang et al. 2018; Chiu et al. 2018). Since we are dealing with sequential data, hence, LSTM was a natural choice as our baseline neural network model. Although the aim of this work is to develop a technique capable of fusing useful information contained in two different modalities, irrespective of their details, still we spent significant compute time to discover the optimal network hyperparameters. This is done through grid-search confined to a reasonable hyperparameter search space. The hyperparameter search space included number of layers in the network, number of neurons in each layer, activation function for each layer, along with the window size p.

Partial auto-correlation of the series was also analyzed to identify association of the current value in the time-series with its lagged version as shown in Fig. 2. As evident from the figure, the series showed strong correlation with its past

[1] http://www.stat.ucdavis.edu/~clarkf/

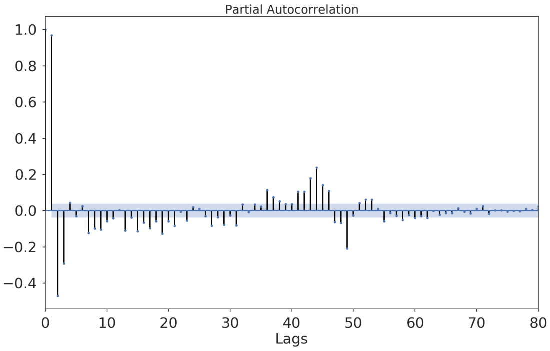


Figure 2: Partial auto-correlation of time-series

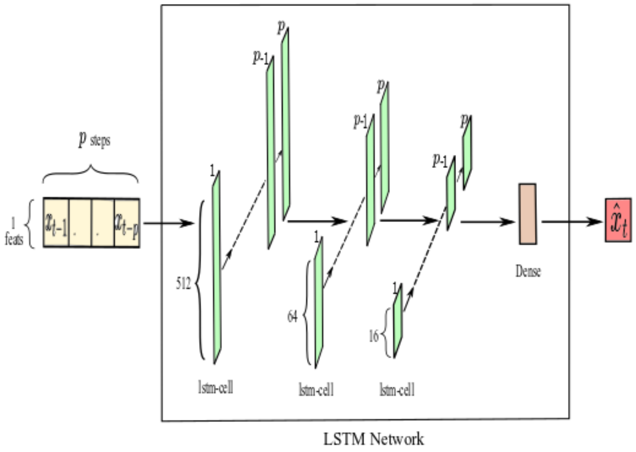


Figure 3: Neural network architecture

three values. This is also cemented by the result of the grid-search that chose the window size of three. The final network consisted of three hidden LSTM layers followed by a dense regression layer. Apart from the first layer, which used sigmoid, Rectified Linear Unit (ReLU) (Glorot, Bordes, and Bengio 2011) was employed as the activation function. Fig. 3 shows the resulting network architecture. The data is segregated into train, validation and test set using 70/10/20 ratio. MSE was employed as the corresponding loss function to be optimized. The network was trained for 600 epochs and the parameters producing the best validation score were used for generating predictions on the test set.

Auto-Regressive Integrated Moving Average (ARIMA) is widely used by experts in time-series modelling and analysis. Therefore, we employed ARIMA as the expert opinion in our experiments. Since the data demonstrated a significant seasonal component, the seasonal variant of ARIMA (SARIMA) was used, whose parameters were estimated using the Box-Jenkins approach (Box et al. 2015). Fig. 4 demonstrates the predictions obtained by employing the LSTM model as well as the expert (SARIMA) model on the test set.

The overall predictions made by both the LSTM as well as the expert network seems plausible as shown in Fig. 4(a). However, it is only through thorough inspection and investigation on a narrower scale that strengths and weaknesses of each of the networks are unveiled as shown in Fig. 4(b). The LSTM tends to capture the overall trend of the data but suffered when predicting small variations in the time-series. SARIMA on the other hand was more accurate in predicting variations in the time-series. In terms of MSE, LSTM model performed considerably worse when compared to the expert model. For this dataset, the discovered LSTM model achieved a MSE of 5.90 compared to 1.24 achieved by SARIMA on the test set.

KINN: Knowledge Integrated Neural Network

Most of the work in the literature (Hu et al. 2016; Gollapalli, Li, and Yang 2017) on incorporating expert knowledge into the neural network focuses on training the network by forcing it to mimic the predictions made by the expert network, ergo updating weights of the network based on the expert's information. However, they do not cater for a scenario where expert network does not contain information about all possible scenarios. Moreover, these hybrid knowledge based network approaches are commonly applied to the classification scenario where output vector of the network corresponds to a probability distribution. This allows KL-divergence to be used as the objective function to be minimized in order to match predictions of the network and the expert network. In case of time-series forecasting, the output of the network is a scalar value instead of a distribution which handicaps most of the prior frameworks proposed in the literature.

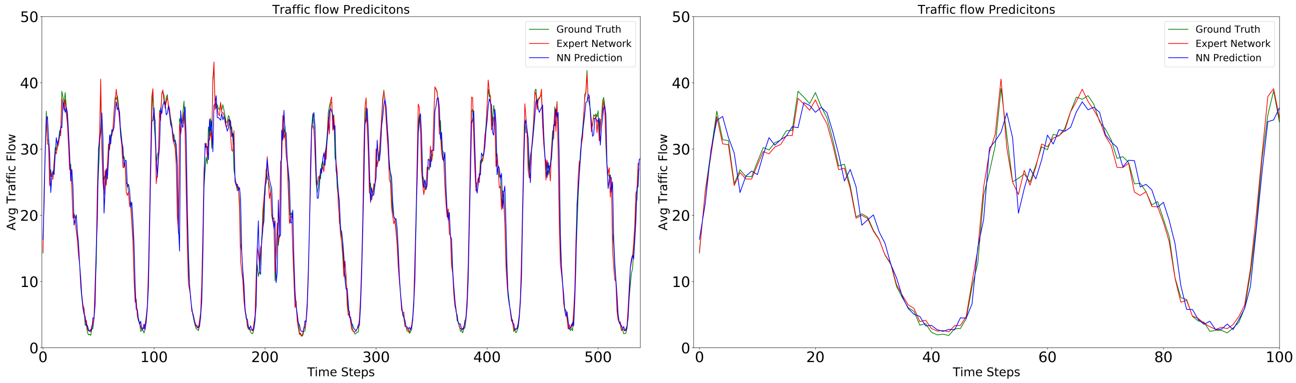
The KINN framework promotes both the expert model as well as the network to complement each other rather than directly mimicking the expert's output. This allows KINN to successfully tackle cases where predictions from the expert are not reliable. Finding the best expert or neural network is not the focus here but instead, the focus is to incorporate expert prediction, may it be flawed, in such a way that the neural network maintains its strengths while incorporating strengths of the expert network.

There are many different ways through which knowledge between an expert and the network can be integrated. Let $\hat{x}_t^p \in \mathbb{R}$ be the prediction made by the expert. We incorporate the knowledge from the expert in a residual scheme inspired by the idea of ResNet curated by (He et al. 2016). Let $\phi : \mathbb{R}^{p+1} \mapsto \mathbb{R}$ define the mapping from the input space to the output space. The learning problem from Eq. 1 after availability of the expert information can be now be written as:

$$\hat{x}_t = \phi([x_{t-1}, x_{t-2}, ..., x_{t-p}, \hat{x}_t^p]; \mathcal{W}) + \hat{x}_t^p$$

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} (x_t - (\phi([x_{t-1}, ..., x_{t-p}, \hat{x}_t^p]; \mathcal{W}) + \hat{x}_t^p))^2 \tag{2}$$

Instead of computing a full input space to output space transform as in Eq. 1, the network instead learns a residual function. This residual function can be considered as a correction



(a) Predictions over the whole test set

(b) Predictions over the first 100 steps

Figure 4: Predictions of NN and Expert Network

term to the prediction made by the expert model. Since the model is learning a correction term for the expert's prediction, it is essential for the model prediction to be conditioned on the expert's prediction as indicated in Eq. 2. There are two simple ways to achieve this conditioning for the LSTM network. The first one is to append the prediction at the end of the sequence as indicated in the equation. Another possibility is to stack a new channel to the input with repeated values for the expert's prediction. The second case makes the optimization problem easier as the network has direct access to the expert's prediction at every time-step. Therefore, results in minor improvements in terms of MSE. The system architecture for KINN is shown in Fig. 5.

Incorporating expert knowledge in this residual fashion serves a very important purpose in our case. In cases where the expert's predictions are inaccurate, the network can generate large offsets in order to compensate for the error, while the network can essentially output zero in cases where the expert's predictions are extremely accurate. With this flexibility built into the system, the system can itself decide its reliance on the expert's predictions.

Evaluation

We curated a range of different experiments each employing KINN in a unique scenario in order to evaluate its performance under varied conditions. We compare KINN results with the expert as well as the DNN in terms of performance to highlight the gains achieved by employing the residual learning scheme. To ensure a fair comparison, all of the preprocessing and LSTM hyperparameters were kept the same when the model was tested in isolation and when integrated as the residual function in KINN.

In the first setting, we tested and compared KINN's performance in the normal case where the expert predictions are accurate and the LSTM is trained on the complete training set available. We present the results from this normal case in experiment # 01. In order to evaluate KINN's performance in cases where the amount of training data available is small or the expert is inaccurate, we established two different sets of experiments starting from the configuration employed in the first experiment. In the first case, we reduced the amount of training data provided to the models for training. We present the findings from this experiment in experiment # 02. In the second case, we reduced the reliability of the expert predictions by injecting random noise. The results from this experiment are summarized in experiment # 03. A direct extension of the last two experiments is to evaluate KINN's performance in cases where both of these conditions hold i.e. the amount of training data is reduced as well as the expert is noisy. We summarize the results for this experiment in experiment # 04. Finally, we evaluated KINN's performance in cases where the expert contained no information. We achieved this using two different ways. We first evaluated the case where the expert always predicted the value of zero. In this case, the target was to evaluate the impact (if any) of introducing the residual learning scheme since the amount of information presented to the LSTM network was exactly the same as the isolated LSTM model in the first experiment. We then tested a more realistic scenario, where the expert model replicated the values from the last time-step of the series. We elaborate the findings from this experiment (for both settings) in experiment # 05.

Experiment # 01: Full training set and accurate expert

We first tested both the LSTM as well as the expert model in isolation in order to precisely capture the impact of introducing the residual learning scheme. KINN demonstrated significant improvements in training dynamics directly from the start. KINN converged faster as compared to the isolated LSTM. As opposed to the isolated LSTM which required more training time (epochs) to converge, KINN normally converged in only one fouth of the epochs taken by the isolated LSTM, which is a significant improvement in terms of the compute time. Apart from the compute time, KINN achieved a MSE of 0.74 on the test set. This is a very significant improvement in comparison to the isolated LSTM model that had a MSE of 5.90. Even compared to the expert model, KINN demonstrated a relative improvement of 40% in terms of MSE. Fig. 6 showcases the predictions made by

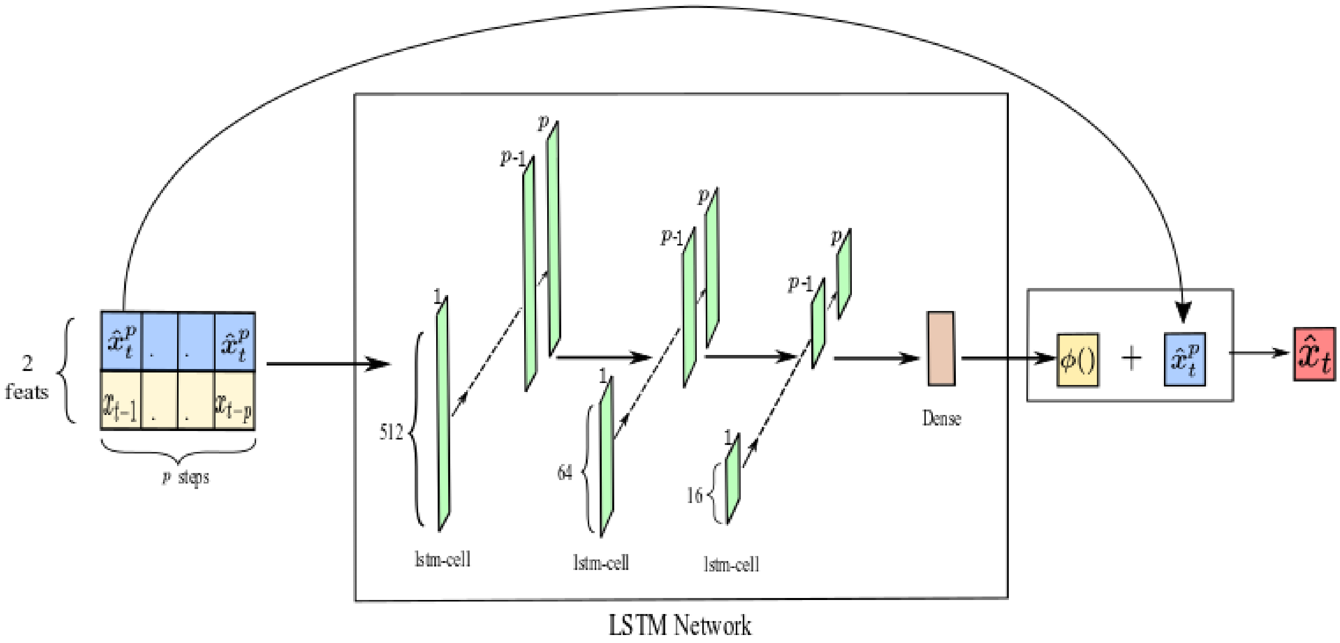
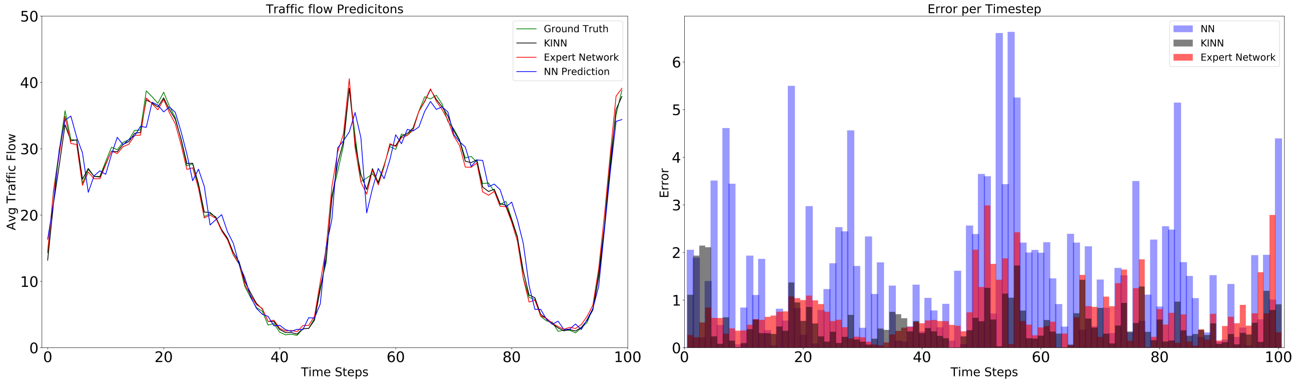


Figure 5: Proposed Architecture

			MSE		
Experiment	**Description**	**% of training data used**	**DNN**	**Expert Network**	**KINN**
1	Full training set and accurate expert	100	5.90	1.24	**0.74**
2	Reduced training set (50%) and accurate expert	50	6.36	1.52	**0.89**
	Reduced training set (10%) and accurate expert	10	6.68	2.67	**1.53**
3	Full training set and noisy expert	100	5.90	7.81	**3.09**
4	Reduced training set and noisy expert	10	6.68	7.81	**3.73**
5	Full training set and Zero expert pred.	100	5.90	621.00	**5.92**
	Full training set and Delayed expert pred.	100	5.90	9.04	**5.91**

Table 1: MSE on the test set for the experiments performed



(a) Predictions of all models

(b) Step-wise error plot

Figure 6: Predictions and the corresponding error plot for the normal case (experiment # 01)

KINN along with the isolated LSTM and the expert network on the test set. It is evident from the figure that KINN caters for the weaknesses of each of the two models involved using the information contained in the other. The resulting predictions are more accurate than the expert network on minimas and also captures the small variations in the series which were missed by the LSTM network.

In order to further evaluate the results, error at each time-step is compared for the isolated models along with KINN. To aid the visualization, step-wise error for first 100 time-steps of the test set is shown in Fig. 6. The plot shows that the step-wise prediction error of KINN is less than both the expert model as well as the LSTM for major portion of the time.

However, there are instances where predictions made by KINN are slightly worse than those of the baseline models. In particular, the prediction error of KINN exceeded the error of the expert network for only 30% of the time-steps and only 22% of the time-steps in case of the LSTM network. Nevertheless, even in those instances, the performance of KINN was still on par with the other models since on 99% of the time-steps, the difference in error is less than 1.5.

Experiment # 02: Reduced training set and accurate expert

One of the objectives of KINN was to reduce dependency of the network on large amount of labelled data. We argue that the proposed model not only utilizes expert knowledge to cater for shortcomings of the network, but also helps in significantly reducing its dependency on the data. To further evaluate this claim, a series of experiments were performed. KINN was trained again from scratch using only 50% of the data in the training set. The test set remained unchanged. Similarly, the LSTM network was also trained with the same 50% subset of the training set.

The LSTM network trained on the 50% subset of the training data attained a MSE of 6.36 which is slightly worse than the MSE of network trained on the whole training set. Minor degradation was also observed in the performance of the expert network which achieved a MSE of 1.52. Despite of this reduction in the dataset size, KINN achieved significantly better results compared to both the LSTM as well as the expert model achieving a MSE of 0.89. Fig 7 visualizes the corresponding prediction and error plots of the models trained on 50% subset of the training data.

We performed the same experiment again with a very drastic reduction in the training dataset size by using only 10% subset of the training data. Fig. 8 visualizes the results from this experiment in the same way, by first plotting the predictions from the models along with the error plot. It is interesting to note that since the LSTM performed considerably poor due to extremely small training set size, the network shifted its focus to the predictions of the expert network and made only minor corrections to it as evident from Fig. 8(a). This highlights KINN's ability to decide its reliance on the expert predictions based on the quality of the information. In terms of the MSE, LSTM model performed the worst. When trained on only the 10% subset of the training set, the LSTM model attained a MSE of 6.68, whereas the expert model achieved MSE of 2.67. KINN on the other hand, still outperformed both of these models and achieved a MSE of 1.53.

Experiment # 03: Full training set and noisy expert

In all of the previous experiments, the expert model was relatively better compared to the LSTM model employed in our experiments. The obtained results highlights KINN's ability to capitalize over the information obtained from the expert model to achieve significant improvements in its prediction. KINN also demonstrated amazing generalization despite of drastic reduction in the amount of training data, highlighting KINN's ability to achieve accurate predictions in low data regimes. However, in conjunction to reducing dependency of the network on data, it is also imperative that the network does not become too dependent on the expert knowledge making it essential to be accurate/perfect. This is usually not catered for in most of the prior work. We believe that the proposed residual scheme enabled the network to handle erroneous expert knowledge efficiently by allowing it to be smart enough to realize weaknesses in the expert network and adjust accordingly. In order to verify KINN's ability to adjust with poor predictions from the expert, we performed another experiment where random noise was injected into the predictions from the expert network. This random noise degraded the reliability of the expert predictions. To achieve this, random noise within one standard deviation of the average traffic flow was added to the expert predictions. As a result, the resulting expert predictions attained a MSE of 7.81 which is considerably poor compared to that of the LSTM (5.90). We then trained KINN using these noisy expert predictions. Fig. 9 visualizes the corresponding prediction and error plots.

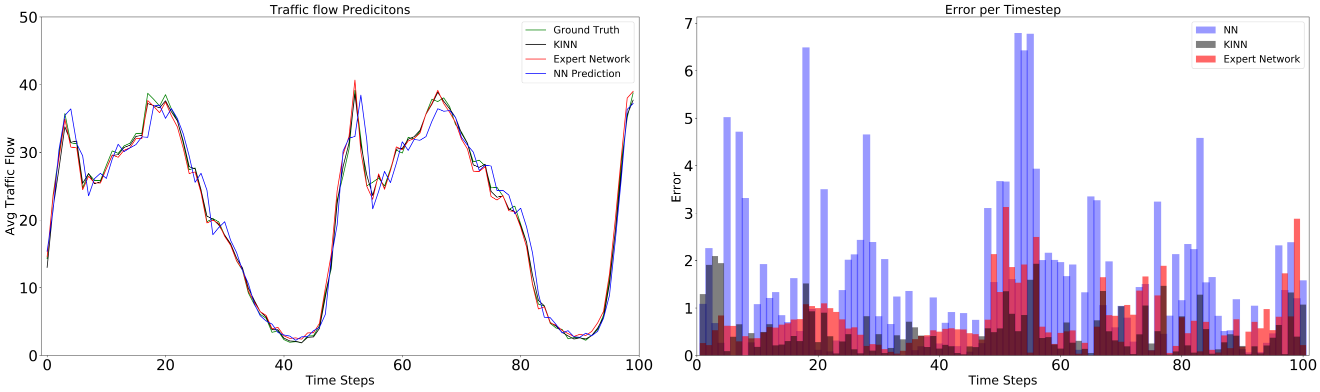
As evident from Fig. 9(a), KINN still outperformed both the expert as well as the LSTM with a MSE of 3.09. Despite the fact that neither the LSTM, nor the expert model was accurate, KINN still managed to squeeze out useful information from both modalities to construct an accurate predictor. This demonstrates true strength of KITNN as it not only reduces dependency of the network on the data but also adapts itself in case of poorly made expert opinions. KINN achieved a significant reduction of 48% in the MSE of the LSTM network by incorporating the noisy expert prediction in the residual learning framework.

Experiment # 04: Reduced training set and noisy expert

As a natural followup to the last two experiments, we introduced both conditions at the same time i.e. reduced training set size and noisy predictions from the expert. The training set was again reduced to 10% subset of the training data for training the model while keeping the testing set intact. Fig. 10 demonstrates that despite this worst condition, KINN still managed to outperform both the LSTM as well as the noisy expert predictions.

Experiment # 05: Full training set and poor expert

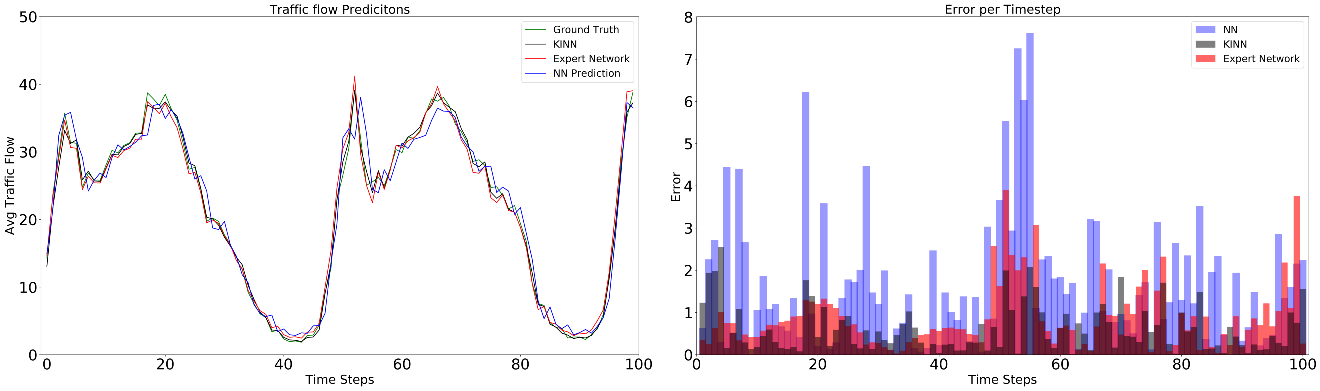
As the final experiment, we evaluated KINN's performance in cases where the expert predictions are not useful at all. We



(a) Predictions of all models

(b) Step wise error plot

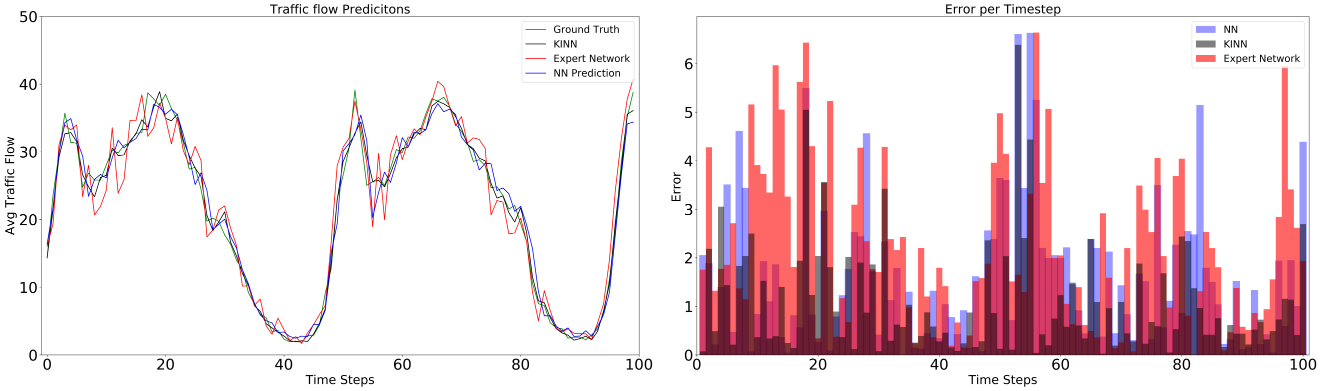
Figure 7: Prediction and error plot with only 50% of the training data being utilized



(a) Predictions of all models

(b) Step wise error plot

Figure 8: Prediction and error plot with only 10% of the training data being utilized



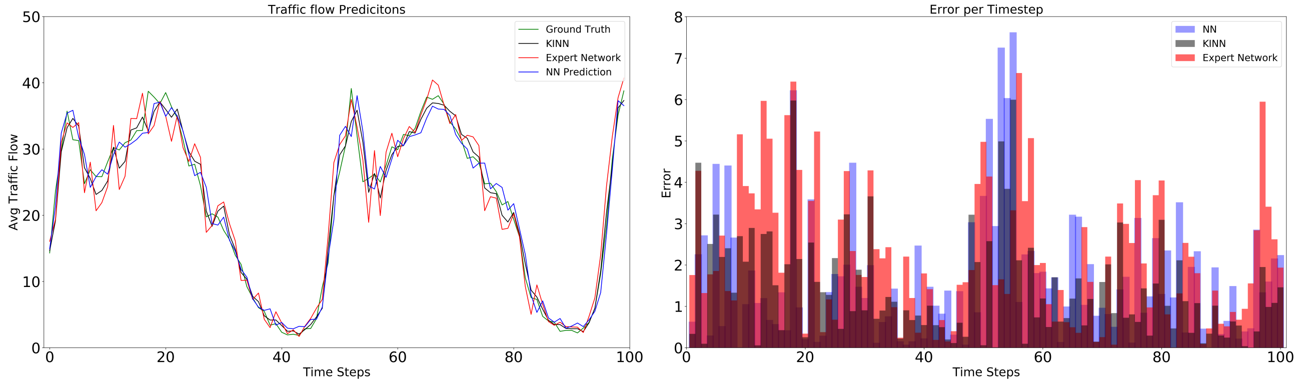
(a) Predictions of all models

(b) Step wise error plot

Figure 9: Prediction and error plot with inaccurate expert prediction

achieved this via two different settings. In the first setting, we considered that the expert network predicts zero every time. In the second setting the expert network was made to lag by a step of one resulting in mismatch of the time step with the predictions. Putting zero in place of $\hat{x}_t^p$ in Eq. 2 yields:

$$\hat{x}_t = \phi([x_{t-1}, x_{t-2}, ..., x_{t-p}, 0]; \mathcal{W}) + 0$$



(a) Predictions of all models

(b) Step wise error plot

Figure 10: Prediction and error plot with inaccurate expert prediction and with only 10% data

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} (x_t - (\phi([x_{t-1}, ..., x_{t-p}, 0]; \mathcal{W}) \\ +0))^2$$

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} (x_t - (\phi([x_{t-1}, ..., x_{t-p}, 0]; \mathcal{W}))^2$$

This is almost equivalent to the normal unconditioned full input to output space projection learning case (Eq. 1) except a zero in the conditioning vector. However, in case of lagged predictions by the expert network, since we stack the expert prediction $\hat{x}_t^p$ in a separate channel, the network assigns a negligible weight to this channel, resulting in exactly the same performance as the normal case.

Table 1 provides the details regarding the results obtained for this experiment. It is clear from the table that in cases where the expert network either gave zero as its predictions or gave lagged predictions, which is useless, the network performance was identical to the normal case since the network learned to ignore the output from the expert. These results highlight that KINN provides a lower bound on the performance based on the performance of the two involved entities: expert model and the network.

Discussion

These thorough experiments advocate that the underlying residual mapping function learned by KINN is successful in combining the network with the prediction made by the expert. Specifically, KINN demonstrated the ability to recognize the quality of the prediction made by both of the base networks and shifted its reliance according to it. In all of the experiments that we have conducted, MSE of the predictions made by KINN never exceeded (disregarding insignificant changes) the MSE of the predictions achieved by the best among the LSTM and the expert model except in case of completely useless expert predictions, where it performed on par with the LSTM network. Table 1 provides a summary of the results obtained from all the different experiments performed. It is interesting to note that even with a huge reduction in the size of the training set, the MSE does not drastically increase as one would expect. This is due to the strong seasonal component present in the dataset. As a result, even with only 10% subset of the training data, the algorithms were able to learn the general pattern exhibited by the sequence. It is only in estimating small variations that these networks faced difficulty when trained on less data.

Conclusion

We propose a new architecture for incorporating expert knowledge into the deep network. It incorporates this expert knowledge in a residual scheme where the network learns a correction term for the predictions made by the expert. The knowledge incorporation scheme introduced by KINN has three key advantages. The first advantage is regarding the relaxation of the requirement for a huge dataset to train the model. The second advantage is regarding the provision of a lower bound on the performance of the resulting classifier since KINN achieves the best of both worlds by combining the two different modalities. The third advantage is its robustness in catering for poor/noisy predictions made by the expert. Through extensive evaluation, we demonstrated that the underlying residual function learned by the network makes the system robust enough to deal with imprecise expert information even in cases where there is a dearth of labelled data. This is because the network does not try to imitate predictions made by the expert network, but instead extracts and combines useful information contained in both of the domains.

Acknowledgements

This work is partially supported by Higher Education Commission (Pakistan), ”Continental Automotive GmbH” and BMBF project DeFuseNN (Grant 01IW17002).

References

Baptista, M.; Sankararaman, S.; de Medeiros, I. P.; Nascimento Jr, C.; Prendinger, H.; and Henriques, E. M. 2018. Forecasting fault events for predictive maintenance using data-driven techniques and arma modeling. *Computers & Industrial Engineering* 115:41–53.

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Chiu, C.-C.; Sainath, T. N.; Wu, Y.; Prabhavalkar, R.; Nguyen, P.; Chen, Z.; Kannan, A.; Weiss, R. J.; Rao, K.; Gonina, E.; et al. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4774–4778. IEEE.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Fildes, R.; Goodwin, P.; and Onkal, D. 2015. Information use in supply chain forecasting.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323.

Gollapalli, S. D.; Li, X.-L.; and Yang, P. 2017. Incorporating expert knowledge into keyphrase extraction. In *AAAI*, 3180–3187.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29(6):82–97.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. 2016. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338.

Pacchin, E.; Gagliardi, F.; Alvisi, S.; Franchini, M.; et al. 2017. A comparison of short-term water demand forecasting models. In *CCWI2017*, 24–24. The University of Sheffield.

Rohrbach, M.; Stark, M.; and Schiele, B. 2011. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1641–1648. IEEE.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature* 529(7587):484.

Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 843–852. IEEE.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Towell, G. G., and Shavlik, J. W. 1994. Knowledge-based artificial neural networks. *Artificial intelligence* 70(1-2):119–165.

Tran, S. N., and Garcez, A. S. d. 2018. Deep logic networks: Inserting and extracting knowledge from deep belief networks. *IEEE transactions on neural networks and learning systems* 29(2):246–258.

Wang, F.-Y.; Zhang, J. J.; Zheng, X.; Wang, X.; Yuan, Y.; Dai, X.; Zhang, J.; and Yang, L. 2016. Where does alphago go: From church-turing thesis to alphago thesis and beyond. *IEEE/CAA Journal of Automatica Sinica* 3(2):113–120.

Weiss, R. J.; Chorowski, J.; Jaitly, N.; Wu, Y.; and Chen, Z. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*.

Wu, Y.; Wu, W.; Li, Z.; and Zhou, M. 2016. Knowledge enhanced hybrid neural network for text matching. *arXiv preprint arXiv:1611.04684*.

Xu, J.; Zhang, Z.; Friedman, T.; Liang, Y.; and Broeck, G. V. d. 2017. A semantic loss function for deep learning with symbolic knowledge. *arXiv preprint arXiv:1711.11157*.

Zhang, X.-Y.; Yin, F.; Zhang, Y.-M.; Liu, C.-L.; and Bengio, Y. 2018. Drawing and recognizing chinese characters with recurrent neural network. *IEEE transactions on pattern analysis and machine intelligence* 40(4):849–862.

Zhao, W. X.; Jiang, J.; Weng, J.; He, J.; Lim, E.-P.; Yan, H.; and Li, X. 2011. Comparing twitter and traditional media using topic models. In *European conference on information retrieval*, 338–349. Springer.