Pattern Classification: Statistical vs. Deep Neural Based Approaches

Josh Reid, Harshwin Venugopal, Sruthy Paul, Ruoxuan Xu
University of Waterloo, Canada
$\{js2reid, hvenugopal, s25paul, ruoxuan.xu\}@uwaterloo.ca$

***Abstract*— This report analyzes the similarities and differences between statistical and deep neural approaches to doing pattern classification. First the history of each technique for pattern classification is presented, then the mathematical background for each is discussed. After this, an example (machine translation) is worked through using both a statistical model and a deep neural network. A detailed analysis is performed on this example and we conclude that each has their benefits and drawbacks in different areas.**

***Keywords* - Deep Neural Networks, Pattern Classification, Statistical Classification, Machine Translation**

I. INTRODUCTION

Pattern classification is a field exploring methods designed to categorize data into distinct classes. A pattern is an object, process or an event. It is an abstraction of some form of repetition found within a set of data. Watanabe[42] defines a pattern "as the opposite of a chaos; it is an entity, vaguely defined, that could be given a name." There are many types of pattern exist, like visual, temporal, sound, logical etc.

Human beings are really good at recognize digits and letters from even a young age. Small characters, large characters, machine printed ones and handwritten ones are all easily recognized. This is taken for granted though when attempting to get a machine to be trained to do the same thing and is an area of intensive study called pattern classification. Pattern classification is the study of how machines observe the environment, learn to recognize patterns of interest from the wide variety of data they collect and attempt to make reasonable decisions about the category of patterns. There are three types of learning algorithms that are used for pattern classification i.e. supervised learning where desired output is known for each input pattern. The second type is unsupervised learning which attempts to find the classes to classify data into without being provided them from the dataset. The third type of training is semi-supervised learning that combines both supervised and unsupervised learning using labeled and unlabeled data for classification purposes. The area of pattern classification has attracted increasing attention due to it having numerous applications that cover a range of important activities such as medical diagnoses, data mining, stock exchange forecast, self driving vehicles and human recognition.

Determining which type of classifier is the most suitable for each pattern classification problem that is encountered in the world can be challenging as it depends on a number of factors such as:

- Hardware available
- Time required to train the classifier
- Time to get results from the trained classifier
- How general the classifier needs to be
- Amount of data available

Pattern classification involving images generally has a large amount of data available, but requires powerful hardware to train a classifier in a reasonable amount of time. Classification of text also has large datasets available and the hardware requirements are much lower, however these generally involve the classification of language which requires a very general classifier to be human readable.

This report will discuss two main scientific approaches to pattern classification: statistical and deep neural networks. In particular, how these two approaches differ from each other which will be further explained through an example that uses both approaches.

Statistical pattern classification is based on the underlying statistical model of pattern and pattern class. Statistical methods are characterized by having an explicit underlying probability model, which provides a probability of a piece of data being in a certain class. There are two main phases of work in the field of statistical pattern classification: the "classical" phase devoted to derivatives of Fisher's early work on linear discrimination and the "modern" phase, which exploits more flexible classes of models, many of which provide a classification rule by estimating the joint distribution of features within each class.

Neural networks process information in a similar manner to that of a biological neuron. The low dependence on domain specific knowledge makes the network standout in the world today. These networks prove to be more efficient and faster to train when compared to the traditional approaches mentioned thus far. These networks are known for the novelty in their architecture as they contain a large number of neurons working together for a specific task. These networks are designed to solve a problem through a phase known as the *Learning Phase*. In this phase the neural networks are trained on the input data. Once the training is completed, there is no human intervention that is required for the networks. Neural networks can identify and classify patterns, and potentially even find patterns that humans may miss because they can consider many different features within data all at once.

The main characteristics of neural networks are their ability to adapt to the data, sequential learning process and their ability to learn more complex input/output features. The most commonly used neural networks for pattern classification

are: feedforward neural networks for supervised learning and Self-Organizing Maps(SOM) for unsupervised learning. In supervised learning, multilayer perceptrons and Radial Basis Function(RBF) neural networks are the most commonly used ones. Figure 3 shows the time for the development of Deep Learning techniques for pattern classification[30].

II. Statistical approaches

In statistical methods, there are two main types of classification: supervised classification and unsupervised classification. The basic characteristics of each has been introduced already. In many statistics literature, these two types often referred to as classification and clustering respectively. The fundamental statistical approaches that are commonly used all belong to supervised learning.

A. Supervised

In statistical pattern classification, a pattern is often represented by a set of *d* features. These are measurable quantities obtained from the patterns and the classification task is based on their respective values. A d-dimensional data vector X_d = (x_1,x_2,...x_d)^T of measurements is used to describe a pattern, where T denotes vector transpose). Therefore, the features are the variables specified by the investigator which is crucial for classification performance. The task at hand is to assign these patterns into one of *c* possible classes with is denoted as ω_i, *i* can be 1,2,3,...*c*. A decision rule is used to partition the measurement space into c regions Ω_i.

The recognition system is operated in two steps: learning and classification. The basic model for statistical pattern recognition is shown in Figure 1. In the training step some preprocessing is first performed to clean the data to help isolate the patterns of interest from the background. Feature extraction finds aspects of the data that represent the pattern of interest and return them as a feature vector which is used to assign the object to a class. The classifier is trained to take this feature vector and classify it as the target output correctly as often as possible. Feedback from the output in the form of an accuracy score allows the software and its designer to optimize the preprocessing and feature extraction methods. Techniques for evaluation of the recognition performance accuracy on training data is needed for designers to find an appropriate point to stop the training, and there are several methods available depending on the type of data being analyzed.

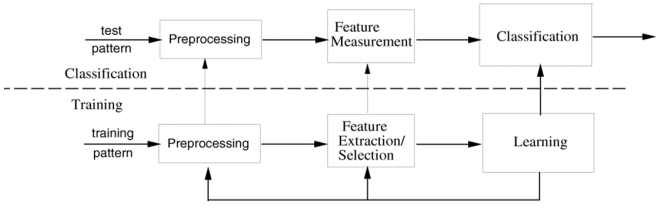

Fig. 1. Model For Statistical Pattern Classification

There are couple decision making rules such as Bayes decision rule, maximum likelihood rule, minimax, Neyman-Pearson and so on. Bayes decision rule is an approaches based on the knowledge of the probability density function of each class. Consider C classes ω_i,...,ω_c, with a priori probabilities P(ω_i),...,P(ω_c). To minimize the error probability, with no extra information, a pattern would be assigned to class ω_j if:
P(ω_j) > P(ω_k), k=1,2,...c; k$\neq$ j.
If there is an observation vector x, considered to be a random variable whose distribution is given by P(x|ω), then assign x to class ω_j if:
P($\omega_j|x$) > P($\omega_k|x$), k=1,2,...c; k$\neq$ j.
The posterior probabilities P($\omega_j|x$) can be expressed in terms of the prior probabilities and the class-conditional density function P($x|\omega_i$) using Bayes' theorem and so the decision rule may be written: assign x to ω_j if :
P($\omega_j|x$)P(ω_j) >P($\omega_k|x$)P(ω_k) , k=1,2,...c; k$\neq$ j.
And this is known as *Bayes decision rule for minimum error*.

Next, consider a different rule that minimizes an expected loss or risk. $\lambda_{ij}|x$ is used to measure the cost of making the decision, a pattern belongs to ω_i, when the true class is ω_j.

Therefore, the *Bayes decision rule for minimum risk* can be expressed as:

$$\sum_{j=1}^{n} \lambda_{ji} P(\omega_j|x)P(x) \leq \sum_{j=1}^{n} \lambda_{jk} P(\omega_j|x)P(x) \ ,$$

k=1,2,...,c

B. Unsupervised

In some sense, most of the approaches in statistical pattern recognition are trying to implement Bayes decision rules. However, since finding labeled data can be difficult and it is very hard to produce, more and more researchers are finding ways to use unlabeled data for classification. Because less information is available to the system designer as a result, the difficulty of doing classification increases. The field of cluster analysis is for dealing with these kinds of decision making problems using unsupervised learning techniques. Cluster analysis algorithms as well as various other kinds of techniques are collectively known as *exploratory data analysis* methods. One of the very popular clustering method used is K-means, which has wide applications.

In the application section of this report machine translation if performed, morphology is concerned with the ways sentences are formed from smaller meaningful units of words. Morphology benefits statistical machine translation in the recognition of unknown words. By defining morphemes in unknown words, their syntactic functions can be inferred and the translation can be performed based on known morphemic components. Learning morphology as a task for natural language does usually come with a set of rules and examples. However, morphology can successfully be learned by a machine if adequate linguistic information and external knowledge is provided. A good example of this is shown by Yarowsky and Wicentowski[48] who use syntactic information and a list of <infection, root> examples to induce

inflectional morphology, both regular and irregular, of unseen words. Ando and Lee [2] proposes a similar multi-gram approach to segment Japanese kanji sequence of compound nouns. They consider a set of n-grams but compare counts of each separately. By taking the distribution of one n-gram at a time and using a "voting" system instead of raw frequency, they eliminate the problems faced by previous researchers where there is an overlap between the higher order n-grams and lower order one. Also, their approach allows for the omission of certain n-grams in their model, which is a great breakthrough in statistical machine translation using unsupervised learning algorithm.

C. Advances in history

Statistical pattern recognition has developed a lot over time and below is the timeline for its history. It started with simple decision tree methods(Hunt et al., 1966), and then advanced with the discovery of Bayesian nets in 1986. In 1995, *Support Vector Machines*(SVM) appeared which is a type of pattern classifier based on statistical learning techniques. SVM has been successfully explored for use with facial recognition[33], optical character recognition, categorization[8] and for information retrieval[18]. The most current technique used one is random forest[4].

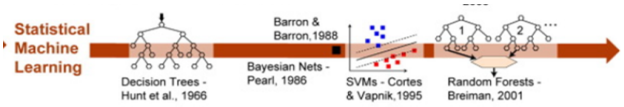

Fig. 2. Statistical Pattern Classification timeline

D. Summary

Nevertheless, within statistical pattern recognition area, significant progress has been made, particularly where the domain overlaps with probability and statistics. And there has been many amazing new developments in methodology and applications built on solid foundations of earlier research and with the modern computational resources available nowadays.

However, there are couple limitations for statistical methods to tackle:
1) The curse of dimensionality - The number of features is too large relative to the number of training samples.
2) Classifiers complexity - The number of unknown parameters associated with the classifier is large.
3) Generation problems - It is very easy to over-train the algorithm which cannot generalize well.
Currently there are new approaches being developed to overcome these challenges such as deep neural networks which are playing an increasingly important role in pattern classification. Statistical pattern recognition techniques are still very popular among certain areas because statistics and probability are excellent tools to deal with noisy data and the models are easier for humans to interpret.

III. Neural Network approaches

The idea of Neural Network emerged from seminal work [1] by logicians Walter Pitts and W.S McCulloch in 1943. They named their model McCulloch-Pitts neurons, which is still considered to be the fundamental model for neurons. The neuron logic proposed by them is known as *threshold logic unit* and was developed to mimic the way biological neurons work. Further in 1950, Alan Turing proposed *The Turing Test* to determine if a human can distinguish a conversation by a machine and a human, although he himself used to call it *Imitation Game*. In his paper [40], he even hints about the genetic algorithm that we use today for neural networks.

It was not later than 1957, the embryo of modern neural network was developed by Frank Rosenblatt. In his paper [12], this psychologist defined a system that could mimic information processing similar to that of human brain and the model was called *Perceptron*. He had visualized his idea more in a hardware sense than in a software sense consisting of algorithms, but it did definitely planted the seeds for what would later grow to become deep neural networks. One of the heavy influenced milestones that inspired artificial neural networks was the discovery of simple cells and complex cells [23] by David H, Hubel and Torsten. In 1960, Professor Bernard Windrow and his graduate student Ted Hoff developed a model based on McCulloch-Pitts neuron. The model was named Adaline [43]. It could adjust weights according to the weighted sum of inputs in the learning phase while in the standard perceptron the net is passed to the activation function and the function's output is used for adjusting weights. The idea of the behavior of these systems from their inputs to how the behavior is modified by feedback was proposed by Henry J. Kelley in Control Theory [25]. This was later used to develop the basics of backpropagation and was used in training neural networks. In 1965, a mathematician Ivakhnenko and associates created first working deep learning network, applying all the theories and ideas that had emerged until then. They named it *Group Method of Data Handling*(GMDH). In this model the learning algorithm used deep feedforward multilayer perceptrons using statistical methods at each layer to find the best features and forward them through the system.

The golden era of neural network is said to have emerged from 1960 till it ended in 1969, when Mavin Minsky, one of the fathers of AI, killed the emerging embryonic idea of neural network stating that the Perceptron model failed to learn simple XOR function. He concluded that the model learned only linear functions and not non-linear functions. This phase that proceeded 1969 is said to be the *First AI Winter*. Just as Minsky could bring the ice, in later years the ice began to melt.

Between 1979 and 1980 , Kunihiko Fukushima developed an ANN model that could recognize visual patterns. The network was named *Neocognition* [13]. This was used in recommendation system at that time. In 1982, John Hopfield created a network and named it *Hopfield Network* [22] which is still remain as a popular implementation even today. It is a

recurrent neural network that serve as a content addressable memory system. In 1985, a computational neuroscientist Terry Sejnowski developed a program *NETtalk*[37] that learns to pronounce English words in a manner similar to that which a child does and was able to improve as time passes. The *First AI Winter* ended in 1986 when Geoff Hinton et al. published a paper[36] *Backpropagation*.

Yann LeCun in 1989, combined convolutional neural networks with backpropagation algorithm which was popular at that time, to develop a model that read handwritten digits[31]. In late 90s and early 2000s, this model was used in United States to read handwritten cash checks and zip codes. He was also the person who developed CNN(convolutional neural network) also known as *LeNet*. In 1989 Christopher Watkins proposed the concept of *Q-Learning*[12] which means learning from delayed rewards. This new algorithm improves the practicality and feasibility of reinforcement learning. The algorithm suggests that it was possible to learn optimally without modeling the transition probabilities or expected rewards. A German scientist Jurgen Schmidhuber solved a *very deep learning* task using recurrent neural networks with 1000 layers. This was a huge leap forward in the complexity and ability of neural networks.

The *Second AI Winter* that started from 1997 was put to end by the development of SVMs in 1995 by Vapnik and Cortes[8]. A SVM is basically a system that could identify similar data and can be used for various applications. Later in 1997, Schmidhuber and Hochreiter proposed a recurrent neural network(RNN) based framework called *Long Short-Term Memory*(LSTM) [21]. This model outperformed the traditional RNN by eliminating the long term dependency problem. Thus LSTM would have greater efficiency and practicality as compared to that of RNN.Further in 1998, LeCun published another paper [32] which was yet another advancement in the field of deep learning. In his paper he proposes stochastic gradient descent algorithm which when combined with backpropagation algorithm outperforms all the existing algorithms.

Launch of *ImageNet*[10] was in 2009 by a professor Fei-Fei Li at AI Lab of Standford University. This was a free database of 14 million labelled images used to train the neural nets for supervised learning. The success of a CNN based creation *AlexNet*, which won several competitions at international level, was a revitalization for the deep learning researchers. It was developed by Alex Krizhevsky in 2011

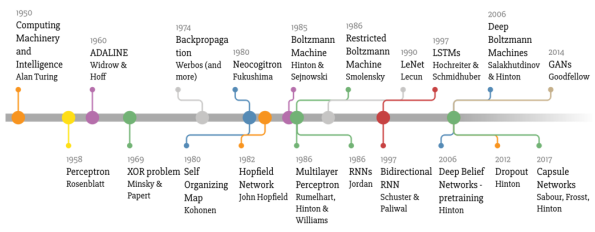

Fig. 3. Deep Learning Timeline

and it outperformed LeNet5. AlexNet consists of 5 convolutional and 3 fully connected layers which strengthens the speed and dropout using *rectified linear units* . Another baby step improve in AI in the case of unsupervised learning was *The Cat Experiment* [1] in 2012. In this experiment a neural network was trained over 1 million unlabeled images of cat and the network outperformed many of the previous attempts of unsupervised learning.

In 2014, Facebook developed and released *DeepFace* [38] which uses several neural networks to identify faces. This model has an improvement of 27% over all the previous reports. This model is said to be a rival for humans with an accuracy of 97.5%. According to LeCun , one of the most interesting idea of Machine Learning in past 10 years is adversarial training also known as *Generative Adversarial Networks*(GANs) [49]. This was introduced by Yann LeCun and researchers in 2014. GANs tackle unsupervised learning problems intelligently. This model consists of 2 nets: Generator and Discriminator, which simultaneously compete against each other and push each other to get smarter and faster. In 2017, was the development of *Capsule Network* [46], which outperformed all the drawbacks of CNNs in the field of image processing.

Today, the world is led by the deep learning and behind-the-scenes of lot of things are neural networks ranging from Netflix and Amazon recommendation engines, Google's voice and image recognition, Apple's Siri and automatic email and text replies, chatbots etc. to a lot of things around us. This field continues to evolve and there is no set timeline for something complex.

A. Neural Networks in Pattern Classification

The first neural network model that was used for Pattern Classification was the Perceptron model in 1989 [16] by P.M Grant. This was used for the concept of linear matched filtering which is used in communication and radar signals. Later in 1992 Won-Hoi Choi et al.[19] applied neural networks to identify patterns to detect fault in induction motor. Nallasamy Mani and Bala Srinivasan [3] were the first to use neural nets for *Optical Character Recognition*(OCR).An artificial neural network based rain attenuation model was first proposed by Chen He et al.[47] in 2000.

In 2006 neural network was first time used in the field of medical imaging by Lin he et.al [20]. Patterns were recognized and classified for electrocardiograms(ECG). Marzuki et al.[35] in 2007 were the first group of researchers to use neural networks for Face Recognition. In the same year Wenjin Dai et.al. [9] used neural networks for improving the accuracy in predicting short term loads by an electric system. Further the same idea was implemented in stock market prediction by Xun Liang et al [17]. Moving ahead, currently all the *Interactive Voice Responses*(IVR) with pattern classification uses neural networks. It was developed by Ali Shah et. al.for the first time in 2009.

IV. Application

In order to tangibly demonstrate the similarities and difference of statistical and neural pattern recognition an example was created for each for Machine Translation(MT). MT is an old problem and the topic has been actively researched to this day in both the the statistical and neural network domains. What exactly is machine translation? It is the translation of text, usually from one language(called parent or source language) to another target language, by a machine with no human association. It is an ideal example for this analysis with a neck to neck comparison of the performance of both the candidates.

Google was using statistical machine translation (SMT) until September 2016. After which it changed its translation system to Neural network based approach called as Google neural machine translation(GNMT)[45]. In quick succession Facebook introduced a new model of neural network, with CNN which gave similar results as Google, but 9 times faster[14][15]. As a reply, Google released their model entirely based on attention mechanism. Even other tech giants like Amazon, Microsoft, IBM had their share in the progress of Machine translation. IBM and John Hopkins university were the forerunner in the field of machine translation, with their initial work on statistical machine translation and extensive research on word alignment[24] in 1980's, which was the major challenge during that time. In the implementation in this report, a basic model in both statistical and neural machine translation is used describing a translation from French to English.

A. Neural machine translation

Neural machine translation (NMT) is a type of machine translation that uses neural networks trained on parallel, bilingual datasets. The early versions of this used feedforward neural networks that had a single hidden layer and updated weights using backpropagation. This updated the weights based on the words that it selected and what the actual words were supposed to be which is determined from the other half of the parallel text dataset. The neural network output a vector for each word in the output phrase of which the largest value was taken as the correct word (softmax). The drawback of these was that the translation of each word in languages is highly dependent upon context, which are the other words around that word. Feedforward neural networks did not put enough weight on the context of each word, so the translation quality was poor.

Recurrent neural networks are a good fit for this job because information is passed between neurons within the hidden layers as words are translated along the sentence, which means they take the context into consideration more for each word in the sentence. This makes them more suitable for machine translation as they take the weight of the previous words in a sentence into consideration as it translates the sentence from beginning to end. This neural network architecture is what was used in this demonstration.

NMT advanced further with the addition of the attention vector that put more emphasis on different parts of the sentence to improve translations. While this has been found to improve the translation ability of NMT [34], it wasn't implemented for this project due to time constraints.

Data Preparation: The dataset contained the bilingual, parallel phrases in one file with the phrases separated by '=>' tokens. So the first step to prepare this data for training is to split the dataset into lines and then split those on this token to create two lists, each with corresponding phrases in each language. Then to remove noise from the text, all the text was converted to only ASCII characters to reduce the variety of words and remove any erroneous symbols. All punctuation and numbers were also removed and each letter was converted to lowercase.

Architecture: The architecture that was used contains two recurrent neural networks in a encoder decoder structure. The encoder converts input text in one language into a constant length word vector that consists of weights that correspond to the known words in the vocabulary of the encoder's tokenizer. The tokenizer converts each unique word into a number since these are faster for the neural network to train on. The number of unique words that the network learns is usually limited to several thousand due to the time of training and amount of data required to learn the weight of each word in a variety of contexts. The decoder then converts this word vector back into integer tokens that correspond to words in the target language. So in this case the pattern being classified is the pattern of words based on the order of text in a different language. The weights correspond to the confidence that a word in the vocabulary is the correct word in the target language at that position in the sentence.

The deep neural network that was made consists of 5 layers, an embedding layer that takes the input, a hidden LSTM layer that performs the encoding, a repeating layer that matches the length from the encoder to the size the decoder takes, another LSTM layer as the decoder and a final dense layer that outputs vectors that are as long as the vocabulary of the target language, which for this example is 4488 words. The number of vectors output for our network is 7, meaning that it can only output phrases that are maximum 7 words long because that is the longest phrase that it trained on. The structure of this neural network is shown visually in IV-A.

Results: This neural network was trained on 22500 parallel phrases and tested on 2500 phrases[26]. The training was evaluated using categorical cross entropy as the loss function to be minimized, which was calculated to be 1.47% error for the final model. This was trained on the dataset for 30 epochs and the overall accuracy of the translation was determined by calculating the BLEU score for the translations in the test dataset. This score measures the percentage of words in the translation output by the machine translator that appear in the actual translation, regardless of their position. While this incorrectly scores translation that make sense but use synonymous words lowly and weighs highly translations that use the correct words but the order doesn't make sense, it is an easy score to calculate quickly that generally reflects translation quality. Higher accuracy error calculation would

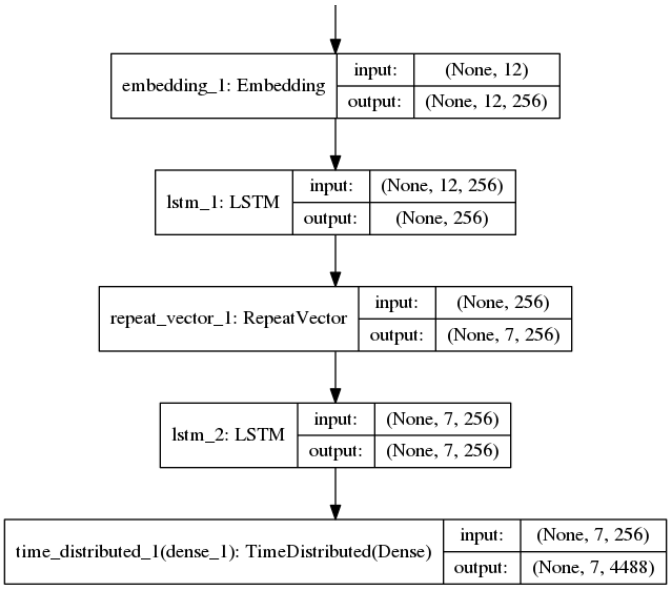

Fig. 4. Structure of the deep neural network used for our neural machine translator

require some natural language processing or require a user that knows both languages to evaluate the quality of the translation manually.

The effect that changing the number of neurons in the LSTM layers and the number of phrases used in the training set was tested, and the results are shown in IV-A. Increasing the number of LSTM neurons caused the accuracy of the translations to improve greatly. The BLEU 1-gram score increasing by 31% and the BLEU 4-gram score increasing by 154% after training on 22,500 phrases for 10 epochs. However, doubling the number of LSTM layers also caused the training time to increase by 246% due to there being a 251% increase in the total number of neurons to be trained. There is a larger increase than just 100% because both the encoder and decoder's number of LSTM neurons are increased, and the input and output layers need more neurons to connect to the increased number of neurons in the LSTM hidden layer.

Changing the number of phrases used to train the neural network had less of an impact on the accuracy of the translations than changing the number of LSTM units. However, in general increasing the number of phrases used to train on improved the quality of the translations. Increasing the number of phrases by 20% caused the BLEU-1 scores to increase by 10%, but caused the training time to increase by 60% and the number of neurons to increase by 9% because increasing the number of lines it trains on also increased the vocabulary size since there was more new words, requiring a larger input layer.

Future Work: In order to further improve the translation accuracy there are several steps that can be taken both in the training of the neural network and also with the data preparation. Currently when preparing the data for the neural network to train on it ignores all punctuation to reduce the number of unique words, however this causes a new problem of creating new words. For example, "don't" is convert to either "dont" which isn't a word, but will now be treated as one. This can be improved by checking each word to see if it is a contraction and then depending on which contraction it is to convert it into its two parent words. Also, any special characters that are not ASCII are removed which is suitable for English datasets, however when working with other any other language this will cause problems with special characters that are crucial to those languages such as letters with accents on them or unique characters (especially a problem for languages like Chinese).

The accuracy of neural networks can be improved with the addition of an attention vector before creating the word vector to emphasize different parts of the sentence that had a higher output accuracy in a previous epoch. Accuracy can also be improve by increasing the number of LSTM neurons in he encoder and decoder layers, however this will require higher computational power or a long time to wait to get the trained network.

B. Statistical machine translation

Statistical machine translation (SMT) is an example of machine translation where a statistical model is used to translate one language to another. This is done by learning parameters from the bilingual text corpora. The work on statistical machine translation has existed since the 1950's with ideas of applying Claude Shannon's information theory. But the pragmatic work started in 1980's and early 1990's by researchers at IBM and various research universities throughout the world[5]

The approach behind SMT comes from information theory, a document is translated according to the conditional proba-

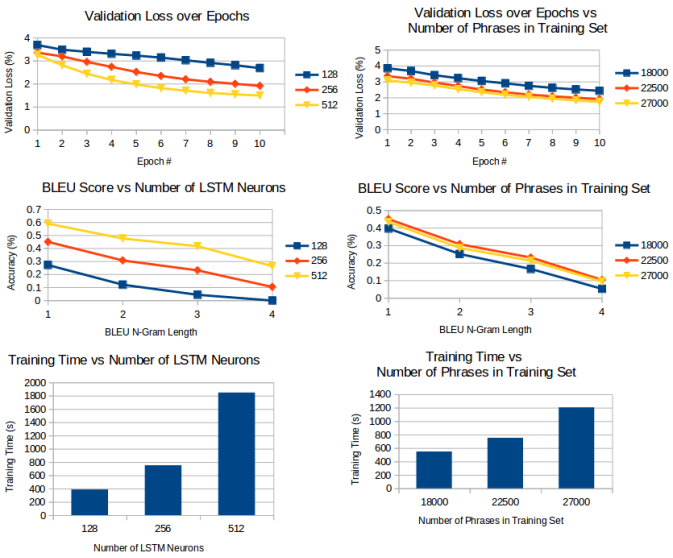

Fig. 5. The effect of training set size and number of LSTM neurons on accuracy and training time

bility distribution $P(e|f)$, where e is the target language and f is the source language. In our example, e is English and f is French. This probability distribution $P(e|f)$ is modeled in numerous ways. One approach which is well suited in terms of computer implementation is *Bayes Theorem* which set forth a proportionality $p(e|f) \propto p(f|e)p(e)$, where $p(e)$ is the language model, i.e is the probability of seeing that target language string and $p(f|e)$ is the probability that the source string is the translation for that target string. The best translation is obtained when the following equation is satisfied.

$$\arg\max_{e \epsilon e*} p(e|f) = \arg\max_{e \epsilon e*} p(f|e)p(e) \quad (1)$$

In a greedy search implementation, one would have to do an exhaustive search by going through all strings(e*) in source language. There are some heuristic approaches to limit this search space, but thats a trade off between the quality and execution time [44]. SMT is a rapidly developing field and it can be broadly categorized into 4 types[44].

- **Word-based** : Fundamental unit of translation is word.
- **Phrase-based** : translates the whole sequence together.
- **Syntax-based** : the idea is to translate syntactic units at a time.
- **Hierarchical phrase-base**: combination of both phrase based and syntax based rules.

Moses - A statistical machine translation system

As a example for statistical machine translation, we try to show the performance of the translation system called as *MOSES*. It is a phrase based machine translation system and often used as a baseline for comparison of machine translation systems. The system is open sourced and well documented with years of support. It was one of the best available statistical systems which achieved results comparable to the most competitive and widely used systems in translation quality and run time. It accounts for all the glamor of Pharaoh decoder described in paper[28]. This also includes a one of new linguistic decoders in SMT, called as confusion network decoding.

Architecture: Architectural organization of Moses can be formulated in four stages:

- Preprocessing - In this phase, the data for training is prepared.
  - **Tokenization**: the corpus is split into words and spaces are inserted between words and punctuation.
  - **Truecasing**: Words in each sentence are converted to the most probable casing so that data sparsity is reduced.
  - **Cleaning**: Long sentence and empty sentences are removed as they will cause problems with the pipeline and it is difficult to align
  - **Word-alignment** : the parallel corpus have to be word aligned so that the mapping probability between the two language can be learned. For this we make use of Hunalign and Giza++ toolkit. Giza++ toolkit makes use of HMM states to align the words and viterbi algorithm to optimize the best states. Venugopal et al[41] propose method and error metric to effectively measure the miss aligned words and Tillman[39] also proposes a further correction of the aligned words.
- Training - In training phase, **Language modeling** is done so that there is fluency in output. The target language, here English is modeled using a 3-gram language model. This is the $p(e)$ in the Bayes' probability equation. Here, phrase extraction, lexical reordering tables are created.
- Decoder - Initial systems of Moses used greedy hill-climbing decoder and produced sufficient results for noun phrases[29]. But current Moses system uses a Beam search decoder with more efficiency.
- Tuning - During decoding Moses uses a linear hypotheses model. Bunch of tuning algorithms are available which can readily be incorporated in Moses, so that an optimal weights for this linear model is obtained, which maximizes the translation performance. MIRA tuning algorithm [7] is one of the latest batch tuning algorithms. After decoding sentences, MIRA updates the weights with respect to some metric.

Results: This section describes the results which we obtained for French to English translation. For the same dataset used for neural machine translation. The preprocessing and training is carried out as described in the previous section. The table I shows the SMT results. Left column is the Source sentence that we feed into our system, middle column is the best obtained results and the final column is the ground truth. We compare the results with the known facts of the Statistical model. The model fails to translate words that are not in corpora. When the statistical anomalies are present, the model does not perform well. An example of a statistical anomaly in this example is when the model is trained on type of distribution, suppose say "news" corpora, and then is tested using text messages(chatbot) corpora, the model does not perform well due to the change in the style of speaking. Also when the word order is different, the translation score is less. The model could not translate idioms, meaning the system is not intelligible yet. The *"UNK"* in the translated result refer to unknown. The model fails to correlate these words while translating.

C. Comparison of NMT and SMT

From our analysis of building the two model there was found to be different areas where each approach was better, which can be extended to other applications of these models. Table II summarizes the comparison of the different areas where each model is better than the other from a usability perspective. It can be seen that training time for NMT is more than SMT, but the time taken to translation after the models are trained is more for SMT than that of NMT. NMT works well even if the data is noisy, but SMT is very sensitive to noise. That is the reason, SMT requires lot of preprocessing in both training phase and while translating. CPU usage is

NMT Result		
French (source)	**English (translated)**	**English (ground truth)**
il y restera vingt ans	hes is over	there will be twenty years
astronomes ligne de temps	drink your	astronomers time line
La vie est amusante	Life is fun	Life is fun
BLEU-1 Score	**45.1%**	

SMT Result		
French (source)	**English (translated)**	**English (ground truth)**
il y restera vingt ans	there will be twenty years	There will be twenty years
astronomes ligne de temps	astronomes |UNK|UNK |UNK line time	astronomers time line
La vie est amusante	Life is fun	Life is fun
BLEU-1 Score	**30.8%**	

TABLE I

more for NMT but less for SMT, this makes it efficient for implementing in mobile devices. When the CPU usage is less, the battery consumption is also less. As these are very big constraints in any mobile devices, SMT earn credits in this aspect. Also SMT is good when there are rare words in the corpus. Neural model can not learn this sparse data, where as the statistical model can work well to some extent with this kind of sparsity. But the Neural method shows its preeminence in far more applications. The neural models can be built to specific applications, where the model works best for that application and it is found to be finest. For example, a trending topic currently in NLP is style transfer. Neural networks can perform this task quite well but statistical models find it very difficult. Neural models learn better feature representations, which can be used for numerous applications. It can also be well suited for multilingual or multi domain translation. More exciting work in this domain of machine translation is by using neural network to translate without using parallel corpora. Thus eliminating the effort of creating the labeled corpus. The research towards this direction can be seen in the paper by Lample et al [30]. Where they convert the text into a different feature space and try to reconstruct the translation by just learning the similar feature representation for the target domain from the target language of random sentences.

Comparison of machine translation systems		
Parameter	**NMT**	**SMT**
Training time	More	Less
Decoding time	Less	More
Training data	Less	More
Space on Disk	Less	More
CPU usage	More	Less
Tolerance to Noisy data	Yes	No
Rare word problem	No	Yes

TABLE II

V. CONCLUSIONS

With all these analysis pertaining to machine translation for both the domain, can we really say, one approach is better than other? Perhaps the reader might be overwhelmed by Neural network performance and be convinced that it can outperform any task. But the competence of Statistical methods are not dreadful. In fact there are cases where statistical models outperform neural in. In the paper [6], Camlica et al shows how local binary patterns with SVMs, provide very good results in medical image classification and retrieval task. But in some cases, it is evident that neural performs better as in the example presented in this report of machine translation. But rather than considering them as a rivalry contest, consider them as a complimentary task for eradicating each other's drawbacks, together they could achieve far more effectiveness in solving problems in a general way. As in the paper by Feng et al. [11], where they train a stochastic neural network to draw samples from target distributions for probabilistic inference. They use a Bayesian inference to learn hyper-parameter adaptively. Also, there are cases where a very big conundrum is solved using statistical learning along with neural. For example the problem of vanishing gradient is solved using KL-divergence, which is a circumstance of Bayes algorithm, which provides additional domain knowledge about the problem [27]. There are good amounts of research, where statistical models are embedded in the optimization stage of neural network. It has also been seen that statistical evaluation metrics are reliable and aid in automatic evaluation without the need of human evaluators, which would not be cost effective otherwise. This paper exhibited the comparative performance of statistical and neural learning with an illustration of their performance in machine translation task, their advancement chronologically and applications. In conclusion, both the domains can be used together to solve far greater problems and each individually has their own domains of pattern classification where they are better by most metrics, but it is a very multifaceted comparison.

ACKNOWLEDGMENT

We would like to thank Professor Karray for his excellent teaching of this course and for providing us a strong fundamentals to several different machine learning techniques. We would also like to thank the TA's Alaa ElKhatib and Chaojie Ou for helping to further our knowledge in the tutorials, and for having the patience to mark all of these reports and final exams in such a short time.

REFERENCES

[1] Teresa M Amabile and Julianna Pillemer. Perspectives on the social psychology of creativity. *The Journal of Creative Behavior*, 46(1):3–15, 2012.

[2] Rie Kubota Ando and Lillian Lee. Mostly-unsupervised statistical segmentation of japanese kanji sequences. *Natural Language Engineering*, 9(2):127–149, 2003.

[3] Jayanta Kumar Basu, Debnath Bhattacharyya, and Tai-hoon Kim. Use of artificial neural network in pattern recognition. *International journal of software engineering and its applications*, 4(2), 2010.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[6] Zehra Camlica, Hamid R Tizhoosh, and Farzad Khalvati. Medical image classification via svm using lbp features from saliency-based folded data. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, pages 128–132. IEEE, 2015.

[7] Colin Cherry and George Foster. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics, 2012.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] Wenjin Dai and Ping Wang. Application of pattern recognition and artificial neural network to load forecasting in electric power system. In *Natural Computation, 2007. ICNC 2007. Third International Conference on*, volume 1, pages 381–385. IEEE, 2007.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[11] Yihao Feng, Dilin Wang, and Qiang Liu. Learning to draw samples with amortized stein variational gradient descent. *arXiv preprint arXiv:1707.06626*, 2017.

[12] R Frank. The perceptron a perceiving and recognizing automaton. *Cornell Aeronautical Laboratory, Buffalo, NY, USA, Tech. Rep*, pages 85–460, 1957.

[13] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (5):826–834, 1983.

[14] Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*, 2016.

[15] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

[16] PM Grant. Artificial neural network and conventional approaches to filtering and pattern recognition. *Electronics & Communication Engineering Journal*, 1(5):225–232, 1989.

[17] Xinyu Guo, Xun Liang, and Xiang Li. A stock pattern recognition algorithm based on neural networks. In *Natural Computation, 2007. ICNC 2007. Third International Conference on*, volume 2, pages 518–522. IEEE, 2007.

[18] Srinivas Gutta, Jeffrey RJ Huang, P Jonathon, and Harry Wechsler. Mixture of experts for classification of gender, ethnic origin, and pose of human faces. *IEEE Transactions on neural networks*, 11(4):948–960, 2000.

[19] Young-Seong Han, Seong-Sik Min, Wo-Ho Choi, and Kyu-Bock Cho. A learning pattern recognition system using neural network for diagnosis and monitoring of aging of electrical motor. In *Industrial Electronics, Control, Instrumentation, and Automation, 1992. Power Electronics and Motion Control., Proceedings of the 1992 International Conference on*, pages 1074–1077. IEEE, 1992.

[20] Lin He, Wensheng Hou, Xiaolin Zhen, and Chenglin Peng. Recognition of ecg patterns using artificial neural network. In *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, volume 2, pages 477–481. IEEE, 2006.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479, 1997.

[22] John J Hopfield, DI Feinstein, and RG Palmer. âĂŸunlearningâĂŹhas a stabilizing effect in collective memories. *Nature*, 304(5922):158, 1983.

[23] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

[24] JHU. Direct models and their application to word-alignment and machine-translation.

[25] Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

[26] Kelly Kelly. Bilingual sentence pairs: Selected sentences from the tatoeba corpus. 2018.

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[28] Philipp Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Conference of the Association for Machine Translation in the Americas*, pages 115–124. Springer, 2004.

[29] Philipp Koehn and Kevin Knight. *Noun phrase translation*. PhD thesis, University of Southern California, 2003.

[30] Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

[31] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[33] Yongmin Li, Shaogang Gong, Jamie Sherrah, and Heather Liddell. Multi-view face detection using support vector machines and eigenspace modelling. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, volume 1, pages 241–244. IEEE, 2000.

[34] Manning Luong, Pham. Effective approaches to attention-based neural machine translation. 2015.

[35] Shahrin Azuan Nazeer, Nazaruddin Omar, Khairol Faisal Jumari, and Marzuki Khalid. Face detecting using artificial neural network approach. In *Modelling & Simulation, 2007. AMS'07. First Asia International Conference on*, pages 394–399. IEEE, 2007.

[36] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[37] Terrence J Sejnowski and Charles R Rosenberg. Parallel networks that learn to pronounce english text. *Complex systems*, 1(1):145–168, 1987.

[38] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[39] Christoph Tillmann. A projection extension algorithm for statistical machine translation. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 1–8. Association for Computational Linguistics, 2003.

[40] Alan Mathison Turing. Computing machinery and intelligence. *Brian Physiology and Psychology*, 213, 1995.

[41] Ashish Venugopal, Stephan Vogel, and Alex Waibel. Effective phrase translation extraction from alignment models. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 319–326. Association for Computational Linguistics, 2003.

[42] Satosi Watanabe. *Pattern recognition: human and mechanical*. John Wiley & Sons, Inc., 1985.

[43] Bernard Widrow et al. *Adaptive" adaline" Neuron Using Chemical" memistors.".* 1960.

[44] Wikipedia contributors. Statistical machine translation — Wikipedia, the free encyclopedia, 2018. [Online; accessed 18-July-2018].

[45] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[46] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.

[47] Hongwei Yang, Chen He, Wentao Song, and Hongwen Zhu. Using artificial neural network approach to predict rain attenuation on earth-space path. In *Antennas and Propagation Society International Symposium, 2000. IEEE*, volume 2, pages 1058–1061. IEEE, 2000.

[48] David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics, 2001.

[49] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.