Term Paper

For Research Directions

University of Nebraska at Omaha

by

Chad Crowe

November 2018

Supervisor

Dr. Hall

Acknowledgements

– To Dr. Hall and her patience and guidance in my research

Contents

List of Figures

List of Tables

0.1 Overview of Applied Machine Learning

Applied machine learning is a large topic. It takes the topics and tools of machine learning and theorizes applications for the real-world. The research explores new ways to represent, manipulate, and guide data analysis with machine learning tools. The research often creates new architectures that perform better in an applied scenario. There are many tools within machine learning. Machine learning architecture is also very flexible. For example, each machine learning model can consist of a varing number of layers, nodes, and even submodels. Moreover, machine learning behavior depends on its input. Some of the papers covered in the related work will explore how data manipulations can further the performance of existing machine learning models. For example, it has been shown that providing regularization produces a model that performs better on new data. Furthermore, there are many ways and techniques to apply regularization. Newer theories have proposed new techniques for regularization. Moreover, many of these new techniques may apply to certain types of machine learning architectures or nodes, such as new forms of regularization for LSTM nodes. Of particular interest to my research has included techniques for handling multi-varied types of input. For example, my masters thesis explored the combination of text and image data within a single machine learning model. These types of advanced machine learning pipelines drive big data, big analysis, and ways to analyze complicated data. I plan on continuing to drive my research within the direction of exploring new ways to handle various inputs with different data types. The field of applied machine learning is vast and full of opportunity for new researchers to discover new theories and make

innovations in the use, application, and understanding of machine learning.

From the related work covered in this paper, there are major threads under development in the arena of applied machine learning. These pertain to machine learning with each type of data. The most popular types of data that receive attention when researchers theorize about their use and possible applications are text, audio, visual, and time-series data. These data types are popular because they are commonly available. Such data are available in public repositories of data or through application APIs, such as text and image data that is available from utilizing the Twitter API. Each data type has multiple facets. More interestingly, certain machine learning algorithms perform better and differently for particular data types. Neural Networks (NNs) perform very well on simple text learning tasks, yet struggle to learn image data. Convolutional Neural Networks (CNNs) may underperform with text-analysis, but have excelled at image-analysis. Video analysis tends to use Long-Term Short Memory nodes (LSTMs), but often incorporate CNNs, since videos are composed of many images.

While there are many techniques for analyzing different types of data, society and academia has seen an increase in the amount of available data. Moreover, not all data is the same. There are many types of video data. There is still video data, moving video data, video at night, day, in different places, and with more people. These data differences exist in text data too. A researcher can analyze text data from Twitter, which looks different than text data from a research paper, which also looks different from text data translated from audio data. The different data types and variations within data types are a challenge for researchers. The creation

of theories about Twitter text data represents advances in our knowledge of social media. Creating models that are better at understanding text data that is created from audible speech can further research's understanding of text sentence structure. These examples only illustrate that there are many ways to approach each data type, and each data type provides many ways to explore and understand its data.

There are many benchmarks within the field of applied machine learning. Many of these benchmarks are the performance of well-tuned machine learning algorithms on a given set of data. It is common for papers within the field of applied machine learning to attempt to improve upon machine learning model performance benchmarks with new tools or techniques. Another common occurrence is to create an architecture that performs well on new data, or with a new aspect of data. The paper will then apply the machine learning architecture against existing benchmarks to show it performs similarly well on the data. It is common to apply the machine learning architecture to existing benchmarks as a litmus test of the machine learning model's overall performance. There are many benchmarks from repositories like Kaggle or libraries like sklearn. New research is always creating new data and creating new benchmarks.

It is worth reviewing the popular tools and approaches within the field of applied machine learning. Machine learning models take inputs. These inputs are transformed by models into outputs. There are many manipulations on the data while it passes through the models. The outputs depend on the type of machine learning. Two common tasks within machine learning are classification and regression. Classification models tend to output a probability for each classification through the final layer,

known as the softmax layer. How the input is manipulated depends on the type of machine learning model. One common type of machine learning model is an Artificial Neural Network (ANNs), which apply simple functions to the inputs, such as a signoid, relu, or tanh function. The model is generally composed of a few layers, where each node is connected to each node in its subsequent layer. The network's input weights, for each layer, are adjusted based on the model's error. The Convolutional Neural Network (CNNs) is similar. The CNN is also composed of nodes. Half of the CNN learning is the same as ANNs, i.e. nodes connect to one another and go through a simple learning function like relu. The difference with CNNs is how the layers are connected. Instead of connecting all layers together, CNNs split the data up into regional chunks, e.g. the upper right array of data. The region of datas are connected. One intersting behavior from the CNN is that region sizes are halved and joined. In this way, regional behavior is captured and joined together in the output. Another major type of machine learning model is a Recurrent Neural Network (RNNs). RNNs are very similar to ANNs, except their layers are connected differently. The RNNs layers do not always connect to the next layer, i.e. the layer's output may serve as input to a previous/the same layer. This provides a cycle of input output information within the neural network. This allows each layer to have some input from future layers, which creates a sense of network memory, where previous data serves as input to the network's current behavior.

0.2 Related Work

Shen et. al explores attention mechanisms for machine learning. The subject of attention mechanisms is not well known within machine learning. It has recently attracted a large amount of attention, due it its performance and speed of computation. Since attention mechanisms are more lightweight, they train faster. The mechanism, as will be explained, still relies on nodes, and therefore has much of the flexibility of neural networks. Shen et. al delve into a type of attention mechanisms, a recurrent attention and bi-way attention mechansism denoted as a Directional Self-Attention Network (DiSAN). The paper shows that its DiSAN model outperforms complicated RNN models in prediction accuracy and time efficiency on existing benchmarks. The paper is relevant because it presents another, quite new machine learning mechanism that has shown promise for applied machine learning.

The attention mechanism takes advantage of a hidden neural network layer. The hidden layer works on the input sequence and predicts the importance of their weights. This creates a mechanism where neural network inputs are scrutinized by a separate neural network. The separate neural network determines the importance of the weights, and give credence to those weights, so that the model primarily uses the most important inputs. The result is a categorical distribution for the input sequence, and the neural network nodes have memory of which input sequences are important or more relevant. One disadvantage of these networks is that temporal order of input information is lost. The paper's DiSAN model helps fix this by providing sequential memory for the attention networks. The paper demonstrates that their attention

mechanism models perform particularly well at alignment scores between two sources, i.e. does well at providing a similarity score between two sources or texts.

There are a few jewels in the Shen et. al paper, like how an additive function for attention often outperforms multiplicative attention, and is also more memory efficient. Their models make us of cross-entropy as an optimization objective and include L2 regularization. The minimization optimizer is Adadelta with mini-batch of size 64. The initial learning rate is quite large, i.e. 0.5, which is decreased over epochs. The weight matrices for networks use GloVe and are pre-trained with out of vocabularty words, which initially were randomely initialized from a uniform distribution. The model uses a dropout of 0.25 and 0.2. The dropout is also varied throughout the learning process. The final model uses fewer parameters than either RNN or CNN networks by margins of 3%. The model is applied to the Standford Sentiment Treebank and performs better than the best existing model by 0.52%. The model is also applied to Sentences Involving Compositional Knowledge (SICK) and with a similar performance. Of important note is the model's bi-directional ability to track different features in forward progressing layers than a backward focused layer, one picking up word families and the latter focusing on word carousel.

Ji Lee performs sequential short-text classification with ANNs. The author's point is that text classifications often occur by only considering a text, not necessarily its preceding or subsequent texts. The paper proposes that using information preceding short texts may improve classification accuracy. Their model initially generates vector representations for short-texts using either the RNN or CNN architectures. The authors utilized early stopping after 10 epochs and performed hyperparameter

training. Their model serves as a benchmark for ANN performance to sequential short-text classification.

Wenpeng et al perform a comparative study between CNN and RNN for Natural Langauge Processing (NLP). This is an interseting subject, since RNNs and CNNs differently model sentences. RNNs capture units in sequence and CNNs are good at extracting positional invariant features. Both CNNs and RNNS are also the primary types of DNNs. The paper covers multiple NLP tasks with each type of network, specifically CNNs, Gated Recurrent Units (GRUs), and LSTMs. It is wroth mentioning that the networks in the study did not obtain great performance on existing benchmarks, which may limit the value of the study's insights. The NLP tasks are sentiment/relation classification, textual entailment, answer selection, question-relation matching, and part-of-speech tagging. The authors found that both CNNs and RNNS provide complementary information on text classification tasks. The authors also found that changing hidden layer sizes and batch sizes resulted in large performance fluctuations. A related work in the study found that RNNs compute a weighted sum of n-grams while CNNs extract the most important n-grams and only consider their resulting activation.

I read a paper on sentence pair scoring by Petr et al. The authors argue that many sentence pairing tasks like Answer Set Selection, Semantic Text Scoring, Next Utterance Ranking, and Recognizing Textual Entailment are all very similar. They propose a unified framework that employs task-independent models for sentence pair scoring models. The model can easily compare models against its baselne in an effort to create a better framework for evaluating machine learning models. It could be

worthy comparing any models I might create for sentence pair scoring within their model framework.

There were a few papers on deep learning with video data. One interesting paper performed deep learning by using CNNs on mulitple frames. This paper was by Hossein Mobahi. The paper performs large-scale object recognition. Videos are composed of multiple frames, which provides a number of frames which contain the same objects. These similar frames can each be processed by a CNN to provide additional information and possibly better accuracy in object recognition. The paper learns the objects, based on the frame-to-frame video motion by performing classification on each frame. The authors see learning from multiple frames more related to evolution, as humans experience learning through the world, which is constantly moving and changing. The paper made use of 72x72 sized images of 100 images, where each object was shot 100 times at angles that each differed by 5 degrees.

Building on the topic of LSTMs within video representations, since many of these interact with both image and motion data. Srivastava et al. created an unsupervised model for learning on video data with LSTMs with the ultimate goal of action recognition. They cite one challenge as tracking multiple objects moving in a background. The paper said that LSTM was useful at extracting and extrapolating motion beyond what the video observed, though that metric seems difficult to measure in terms of goodness or performance. The authors also took the approach of skip-gram models of trying to predict in-between frames to train their model. The final model predicted up to 13 frames in the future and took 20 hours to converge on only 300 hours of data. The resulting predictions and reconstructions were blurry. The blurriness was fixed

by adding more LSTM units to remember image data. They faced the issue of LSTM and their gradients vanishing. Despite this, they had 74.3% accuracy on recognizing actions from video data, which seems respectible. The authors found that the model often loss the ability to keep precise object features in future frames, though it could recreate long-term object motion.

Wojciech Zaremba explores RNN regularization. Dropout is the most successful technique for regularizing neural networks, but they do not work well with RNNS and Long Short-Term Memory units (LSTMs). Dropout works by randomly dropping outputs on a certain percentage of nodes. Dropout is used as a form of regularization to make networks more generic and stable on new inputs. Being able to apply regularization to RNNs or LSTMs could make video deep learning much more performant. Due to lack of regularization effectiveness in RNNs, RNNs tend to quickly overfit on large networks. The author's trick is to apply the dropout operator only to the non-recurrent connections. The final model uses minibatch of size 20 with 650 units per layer of the LSTM.

Graves from Google researched speech recognition with RNNs. The paper presents a system to transcribe audio data to text. The paper is novel because it performs transcription without a phonetic representation. The model uses a bidirectional LSTM RNN architecture. Current practice working from audio to text data comprises speaker normalization and vocal tract length normalization. The normalized voice is then fed into a model. The extra step of normalization of voices makes the model bad at dealing with outlier voices, such as the elderly. By having the model direclty transcript the audio data, the model can overcome the difficulties of odd

voice tracts. There are papers by Graves that perform raw speech with RNNs and Restricted Boltzman Machines (RBMs), but the model is expensive and tends to be worse than conventional processing. A previous work did speech recognition with this architecture (Eybern et al., 2009), but it used a shallow architecture and did not deliver compelling results. An advantage of this research is its use of bidirectional RNNs to capture the whole utterances and their context. This is possibly useful in the core research proposed by the term paper as a method for analyzing the entire context of speech data.

Karen et al. proposes a unique use of combining CNNs for action recognition in videos. The goal is to capture complementary information from still frames and their motion. The recognition of human actions in video is well researched. This paper builds upon existing works by creating a new architecture for analyzing human actions by combining an image-based model and a movement-based model, e.g. one model tracks the still images and the other tracks the gradient of movement in those images. In this way, the paper presents a technique to combine two different data types into a single CNN model. The overall idea is the motion and still-frame data types are very different. Also, actions often contain motion. The action motion can aid to the identification of the action. The CNNs are separately trained and later combined via their softmax scores. The paper does some interesting calculations for the motion in order to account for a moving camera by subtracting movement that exists across the entire frame. The CNNs do max-pooling with a 3x3 window and a stride of 2, which seemed practically too large. The images were 224x224 and randomely cropped, horizontally flipped, and underwent RGB jittering. The authors

found that such fine-tuning only gave marginal improvements over the training set. The paper also saw that large dropout over-regularises learning and leads to a worse overall accuracy.

Translating images to videos is a popular topic. It is particularly interesting because it performs translation from one data type to another. Donahue et al presents a model transcribing video to text descriptions. The paper proposes an architecture known as Long-term Recurrent Convolutional Networks (LRCNs). The goal of the architecture is to levarage CNN recognition strengths and to have an RNN remember time-varying inputs and outputs. One of the largest difficulties with the model is deciding how much time-varying information to remember, which were not deterministic in this model due to their issue of the vanishing gradient in their RNN model. The authors see their model as an improvement on video activity datasets with complex time dynamics and improve existing benchamrks by 4%. The LRCN predicts the video class at each time step and average the predictions for a final classification. The model extracts from 16 frame clips and uses a stride of 8 frames from each video. The model is trained on TACoS, a dataset for video/sentence pairs.

Ji et al. presents another method for recognizing human action with CNNs. The topic is particularly interesting because it represents a model that translates between very different data types. The approach is simple and uses a CNN to predict actions at the frame level. Another CNN then takes inputs from contiguous frames via their location. The end-result is a 3D CNN that combines each 2D frame and uses time as the third dimension. The hope is that the CNN models will capture temporal information from the adjacent frames. This seems like a very good method

to represent the data with very little bias. The model outperformed 2D CNNs, which seems sensible since the 3D model contains the 2D model plus more information.

Alex Graves at Toronto persents a paper on predicting future handwriting using LSTMs. The resulting system was able to generate highly realistic cursive handwriting in a wide variety of styles. The topic is interesting because it is an example of using a different machine learning architecture to translate data from video to another format, i.e. predicted handwriting sequences. The topic of translating data to a type of prediction is interesting for the term paper's topic. Existing work has used LSTMs to generate future sequences in domains like music. RNNs are nice because they are fuzzy in the sense that they do not use exact templates from the training data to make predictions but use internal representations to interpolate from the training data to a result. This RNN reconstitution of training data is an interesting way to transform data from one type to another. It might be interesting to use RNNs as a translator from one data format to another, then use that representation to train another machine learning model. The paper builds on this principle and finds that a better data type translation occurs when the LSTMs are given longer memories. The model uses skip connections to all input layers, which does not connect top RNN layers to bottom RNN layers, assuming that these connections are likely unrelated and unhelpful to the final model. The paper also constrained its gradients to a smaller range to prevent large derivatives in the backpropagation. The authors also found that retraining with iteratively increased regularization results in faster training than random weights with regularization. This makes sense, since the initial weights are likely better than random weights. Their network only took four epochs to converge.

It is good to know that LSTMs can converge so quickly.

Building on learning data type representations, Cho et al. builds upon phrase representations using RNN encode-decoders with the purpose of language modeling. The authors use two RNNs as an encoder-decoder pair. There is definitely an emerging trend in the related work where RNNs are used to create internal data representations for encoding data to another data type. The model translates from English to French and learns the translation probabilty of an English phrase corresponding to a French phrase. The model can conversly be used to score a given pair of input and output sequences. The authors also acknowledged that simply training statistical models do not necessarily lead to the optimal performance.

One paper my Simonyan and Zisserman explore very deep CNNs on large-scale image recognition. The authors found that the trick to having deep CNNs is to have small filters. The authors had 3x3 CNN filters with 16-19 layers and their model are two of the best performing convnets publically available. The authors performed no fine-tuning and fed their model fixed-size 224x224 RGB images. The only preprocessing done is subtracting the mean RGB value computed on the training set from each pixel so that the pixel intensity values tend to fall in a semi-normal distribution around zero. Surpisingly, the stride is kept at one, which is common in practice but requires more time for training. Spatial pooling ocurs five times, which follow the conv layers, though sometimes the model has multiple conv layers before the data is pooled. Max-pooling is performed on a small 2x2 window with a stride of 2, which likely saves a lot of model training time. The authors used a ReLU on all layers and none of the layers contain Local Response Normalization (LRN), as such

normalization only leads to great memory consumption and computation time. The benefit to the many small filters is that the number of weights in the convnet is not greater than that of a shallow net with larger convnet layers. The model utilized a mini-batch of 256 with momentum set to 0.9, which is a common pattern in many of these papers. The model did include L2 normalization with the multiplier set to 5*10-4. A dropout of 0.5 was used, which seems very large compared with most papers, which probably use 0.3. The learning rate is initially set to 0.1 and decreased by a factor of 10 when the validation set accuracy stopped improving. A common theme in the related work is a large learning rate to initialize weights, which then dramatically decrease for small performance increases. The authors obtains fixed-size 224x224 convnet images by randomly cropping and rescaling training images. The crops also underwent random horizontal flipping and random RGB color shifts. The authors emphasized that using a large set of crops can lead to an improved accuracy. The training time took 2-3 weeks and training with 1000 classes on 1.3M images, and tested on 100k images, and validated on 50k images.

0.3 What area within IT would you like to draw upon to inform a topic that you are interested in?

0.4 Topic

0.4.0.1 What is known about this topic?

Little has been invested in developing models for multiple data types. It is less common to have models composed of multiple data types. Moreover, these models can be analyzed separately with good results. Most of existing research concerns translating one data type to another, new applications of existing architectures, or architectures for improving upon curent methods at analyzing one data type. Existing tools allow for combining machine learning models. The libraries and research gap provide opportunity to explore architectures for learning with compound datasets.

0.4.0.2 What needs to be known?

Researcher's need to understand what are methods for capturing related information between text and image data. Research will benefit from knowing if certain architectures are better at capturing correlations between image and text data, even if the final model does not perform better. Such knowledge can be used in conjuction with other models to improve overall performance. The field also needs to know if image and text data are related, if at all, or in which situations are the data related.

0.4.0.3 What would you like to know?

I would like to know if LSTMs are capable of creating internal representations of each data type, so that LSTM models from each model can be combined. I am curious if attention models can be used with single hidden layers to specify which data relationships between text and image data are most important/significant. I am also curious how combining both text and image data in a CNN affects the data learned in each layer. It would also be interesting to train a model on one data type, and then fix the top layers are retrain lower layers on the other data type to see if this gives performance benefits. Such a fixed top/trained bottom data set might gain accuracy from learning some of the information from the other data set. It would also be interesting to iteratively build a combined model based on two individual models, such as building a third combined model using weights from each independent image-CNN and text-NN model.

0.4.0.4 Why is it important?

Applied machine learning is always applying known methods to new data, on new architectures, and in new contexts. Moreover, data is trending towards larger data sets and more diverse types of data. The ability to better analyze a combination of text and image data will make research's approach to combined data types better. This can improve how the field approaches more complex data types and encourage paritcular machine learning architectures or approaches to the data.

0.4.0.5 Why we should care?

Machine learning has many applications that contain a mix of text and image data. For example, a patient's list of symptoms and X-Rays is an example of combined text and image data that this type of research affects. Also, advertisements contain image and text data that is presented to the user. Better methods for analyzing this data can improve what researcher's know about each field.

0.5 Describe your understanding of one or two open problems/questions in IT

0.5.1 What is the gap between what is known and what needs to be known?

What is known is how to individually analyze data types with a great degree of accuracy. There are also known techniques for translating data types to other types, such as transcribing actions in videos to text, or object recognition in images. Data scientists have had success finding correlations between text and image data. There are a few existing approaches. One such approach is to extract features from each data type. These features can then be fed into a NN that can find patterns within each data type. Yet, extracting these features is complicated. Features can be manually created or found by CNN or NN. Yet, running data through a CNN or NN is time intensive and doesn't always result in meaningful featurs. Feeding unmeaningful features into a third model might work, but it might not. Moreover, extracting the initial features requires training a model on outputs. Each trained model is going to be biased to sense data type specific nuances, since that is how each model was trained. The network might have poor weights for sensing interactions between data types. All these situations are likely to lead to a poor result when combingin the outputs of different machine learning models.

While combining models sounds simple, there are many ways to combine the models. The models can be combined at the final layer, the n-1 layer, the n-1 layer etc, or even have their inputs run through the same model until the model converges. It is a difficult scenario and deserves research. This area of research is newer and opportune for exploration by a new researcher.

0.5.2 What would you like to know?

The reserach goal is to capture interplaying features between text and image data. Moreover, the research wants to investigate architectures that are best for handling compound inputs that comprise text and image data. The research will also explore transformations that will aid in the overall analysis of both data types.

0.5.3 Formulate a question that specifies b.

This section formulates a question for the research gap that can be addressed. There are multiple research questions.

Overarching research question: When modeling complex data that is comprised of text and image data, can architectures that combine text and image data discover new features that are useful for existing models. These features are deemed useful if they can improve the final model's performance.

1. Can text and image data be trained on the same architecture to produce significant features for existing models. 2. Based on the pattern of

1. Can compound machine learning architectures outperform find correlations between text and image data which are not captured by individual models on existing machine learning benchmarks

2. Is it possible to extract correlations between text and image data

3. Is there a better way to preprocess image and text data to capture correlations between the dataa. Specifically, will transforming text and image data into LSTMs make the data easier to process by traditional machine learning models

4. How does a CNN model act when trained on both image and text data

5. How does a NN model act when trained on both image and text data

6. Does combining the hidden layers from individual text and image-based attention models create a combined model that better detects correlations between image and text models. a. Can a third combined model be created based on the weights from two individual models.

8. Can LSTMs translate from text to images, and vice-versa to draw out correlated features.

0.6 Research Question

0.6.1 Open Research Problems in IT

0.7 Purpose Statement

0.7.1 Signpost that establishes the central intent for the study?

My thesis was two-fold, initially it was creating simple neural network and convolutional neural networks in order to perform prediction on advertisements. Each network regressed and output that represented how well the advertisement performed on social media. The second goal of my thesis was to somehow combine my analysis of NN and CNNs. The initial ideas were to average their regressed outputs, as an average equally weights outputs from both models. Another idea was to feed each output into a new machine learning model. The outputs from the NN and CNN were fed into a decision tree, hoping that the decision tree would create a result that performed better than simple averaging. The decision tree was slightly beneficial, but overall the combination of the two models performed worse than each individual model. The result denoted that there was no obvious or simple way to combine machine learning models, especially models that were comprised of different data types, e.g. images and text data.

The really interesting and fun part of the thesis occurred while exploring ways to combine the NN and CNN models. The decided upon plan-of-action was to try another method for combining the NN and CNN. The thesis attempted to combine the the NN and CNN mid-model. The important observation was that most of the model's computations were finished before the model's final output layer. The model's could be combined at the n-1 layer. This allowed each model to stay mostly independent, while combining their data. The result was a final model that performed better than

the NN or CNN model. The thesis found that combining models of different data types is interesting, difficult, and worth exploring. The topic of combining models of different data types offers a lot of promise. If researcher's can find better ways to model complex data, i.e. data that contains multiple data types, then researcher's and practitioners will have better tools for analyzing real-world data. Most data in the world is complex, interconnected, and can be represented in many ways. For example, medical diagnosing patients makes use of visual data, e.g. X-Rays and a set of symptons, which can be represented as text data. Having ways to combine the analysis of text and image data can produce models that make better predictions on these datasets.

The central intent for my research is developing methods and machine learning architectures that best process a combination of image and text inputs. The research will investigate ways to manipulate inputs and formulate architectures to handle these types of data inputs. Central to many of the related works were architectures and methods to process the data differently. Current reserach is full of methods and architectures for processing data. It is feasible that some of these methods can be combined or leveraged to work well on a combination of text and image data.

One such interesting method for combining the analysis of image and text data is to make use of attention from machine learning models to remember which combinations of text and image data are most important. The method begins with two shallow models, a text-based model and image-based model. The models are attention models and track which input sequences are most important. A third combination model can be built using the two independent models. The third model can use the

attention layers from each models. In this way, the new model looks for the most important input sequences for the text-based and image-based models. It seems like the one hidden layer that makes use of the text-based and image-based hidden layers might be robust enough to create a combined model that performs well on the data by paying special attention to the most important input features. It would be interesting to see how the behavior of the model changes if another hidden layer is added to the model, sicne this would be combining the most important inputs from each model. This combination of the models is similar to the thesis methodology, which worked well.

Another method is to iteratively build the combined model. I propose that text-based and image-based models are built. I propose that a combined model begins with the inputs of the text and image-based data. For each new layer of the combined model, I propose that the hidden layers and their weights from the text and image-based models are added to the combined model. This allows the combined model to garner important data from each individual text/image-based model while analyzing their combination within the main layer. The added layers can be frozen so that their weights don't change with training. Only the original layers, which are not from the text/image based models undergo training. This allows the model to train on the relationships between the image and text relations in the model.

A third interesting method would be to train a single network twice. CNNs have had some success with text. It would be interesting to train a CNN model on images, freeze the top few layers, and then train the rest of the lower layers on text data. One challenge is formatting the text and image data for a single model. The result might

be useful for finding the major image features and increasing some of the accuracy with text data.

0.7.2 Is there a body of knowledge in IT you would like to contribute to?

The body of knowledge I would like to contribute to is a better analysis of complex data types, specifically analyzing data that contains image and text data. The task of analyzing complex data types is difficult because the complex data types interact differently, depending on their context and how they are used. Moreover, because text and speech are different, a single machine learning model or architecture will introduce different biases on each data type, or will perform worse on one data type than the other. Many decisions must be made before a researcher can begin any analysis on text and speech data. Current practice generally analyzes each data type separately and combine their results. It is also common to transform one data type into the other for ease of processing, such as transcribing a picture into text and descriptive sentences. The body of knowledge in IT would greatly benefit from research into new ways to analyze text and image data. For example, these data types can be mixed into a single architecture that is better at analyzing the combination of data. Another idea is to create two separate models and a combined model that detects interractions between image and text data. The subject alone of finding interractions between text and image data is a great field for adding reserach contributions.

I think the interests I can make research contributions to include complex data

type preprocessing and both model architecture and combination of complex data types. The research would focus on ways to build models that work on a combination of image and text data. Many of the existing tools like attention models and the flexibility of CNN and NN models should go a long way to making a more combined analysis possible. One challenge is the contex where the complex data analysis occurs. Hopefully a combined model analysis and architecture will be general enough to apply to many context. With the need of a general solution is a specific place to begin working on complex machine learning models. One booming industry is the advertisement industry, and most ads are a combination of image and text data. This field seems well suited to begin generating models for working with complex data types.

0.7.3 Is there an area of practice you would like to improve?

I work as a software engineer at LinkedIn on their advertisement reporting team. Our team is responsible for the real-time reporting of advertising events that occur on LinkedIn. Anytime an advertisement is seen or interacted with, our team collects those events and reports them to advertisers. A benefit of being on this team is that we are often involved in the process of creating new advertising platforms and analyzing the performance of advertising campaigns. Currently, I am the leading developer for an advertising strategy where advertisers can target the decision makers at companys who buy their products. Due to my involvement with advertising and analyzing campaign performance, I have particular interest at analyzing advertiser

content. Since our team reports advertising content, we often deal with all the types of advertisements and their metrics, e.g. percentage of video watched by a LinkedIn member, clicks, impressions, and if they click on advertisements or later visit particular webpages via cookies setup by advertisers on LinkedIn. This gives me particular interest in these advertising metrics, whose inputs are diverse types of data, e.g. text, image, and video data.

0.8 Theory

0.8.1 Philosophical Assumptions

0.8.1.1 Social Reality

The overarching theory behind the research question proposes that when multiple data types describe a phenomenon, that those data types are related. The theory is that some of the data from each data type is correlated, since the data is extracted from the same event. For example, social media advertisement contains an image and a description. The user's reaction to an advertisement is likely a factor of not only the image or the description, but a factor of both the image and its description. The theory poses that when data naturally occur together, each type of data contains correlations with the other type of data.

Another theory is that the correlations between data types can be discovered and measured. Someties, even when correlations exist, they may be difficult to discover or even measure those correlations. The theory supporting the research question proposes that these correlations can be measured within machine learning models.

The research question also assumes that new machine learning architectures can better learn when designed to take advantage of correlations that may exist between the different data types. These different architectures can either add data to existing models or improve upon existing models that usually only operate on one type of data. By considering how each data type might interact, and how the model can learn from this interaction, the model may be able to better learn and represent the data and its features.

0.8.1.2 What we understand to be true

Some of the required theories are generally accepted. It is known when multiple data types exist, machine learning models can learn by including both data types in its models. The difference is that usually these data types are separately modeled, or each data type is transformed into a set of features which may be combined. The existing theory performs well. Yet, it is difficult to draw out features from complicated, this may be a shortcoming of current practice.

Current theories have shown that different data types can be analyzed with the same architecture. For example, a CNN can perform image analysis and NLP tasks. Yet, CNNs will perform worse on and differently on NN than on CNNs. Also, a NN can process text and image data, but NN poorly draw out image features. Existing studies have also transcribed one data type to another, such as sentence descriptions of images. Yet, sentence description is generally the goal, not a transformation of data for more analysis.

0.9 How do we understand?

0.10 Means of Investigation

0.10.0.1 Qualitative/Inductive

The qualitative aspect of the research asks for qualitative features the new models can produce. An example of a qualitative feature could be how including text data in a CNN might change the features picked up by a CNN from layer to layer. Such qualitative analysis of CNN layers is common and helps researcher's understand which features each layer is targeting.

A more interesting way of performing qualitative analysis is to create auto-encoders with the new compound architectures. The models can be trained on a combination of text and image data. The models can then generate new images and text based on the inputs, and the results can be qualitatively examined to for general accuracy and patterns that the autoencoder learned from the data.

0.10.0.2 Quantitative/Deductive

The means of investigation are creating models that have promise for processing both image and text data. These models can be trained and evaluated against existing benchmark datasets. The performance of these models validates if the general theories work. The deductive step assumes if the proposed architectures and methods perform well on existing benchmarks, then they will perform well on other data.

0.10.1 Means of Evidence

0.10.1.1 Data Collection

Data collection is not necessary for this analysis. Actually, it is preferred to use existing data and benchmarks for evaluating the machine learning models. Beyond the simple analysis, there are existing datasets for advertisement data. If the study finds performant models, these can be trained on advertising data.

0.10.1.2 Data Analysis

Data analysis for this research is standard. The proposed models and method for preprocessing are trained on known datasets and evaluated on validation sets. The performance of the models shows their overall performance.

The research questions concerns detecting interplay between each data type. This type of analysis is more qualitative. The qualitative analysis may find features which show promise for certain architectures. Such architecture findings add to the existing research.

Bibliography

Baudi, P. and Pichl, J. Sentence Pair Scoring : Towards Unified Framework. (C).

Crowe, C. (2018). Initial Related Work. pages 1–4.

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T., Austin, U. T., Lowell, U., and Berkeley, U. C. Long-term Recurrent Convolutional Networks for Visual Recognition and Description.

Enerative, F. O. R. G., Of, M. O., and Ideos, N. A. V. (2014). V (l) m : a b f g m n v. pages 1–15.

Fan, Y., Lu, X., Li, D., and Liu, Y. (2016). Video-Based Emotion Recognition using CNN-RNN and C3D Hybrid Networks. (November).

Ghosh, A. and Veale, T. (2016). Fracking Sarcasm using Neural Network. pages 161–169.

Graves, A. Generating Sequences With Recurrent Neural Networks. pages 1–43.

Graves, A. (2014). Towards End-to-End Speech Recognition with Recurrent Neural Networks. 32.

Hurri, J. (2003). Simple-Cell-Like Receptive Fields Maximize Temporal. 691(3):663–691.

Ji, S. and Yu, K. (2010). 3D Convolutional Neural Networks for Human Action Recognition.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-fei, L. (2015). Large-scale Video Classification with Convolutional Neural Networks Presenter : Esha Uboweja Problem Classification of videos in sports datasets. (June 2014).

Lan, Z., Lin, M., Li, X., Hauptmann, A. G., and Raj, B. Beyond Gaussian Pyramid : Multi-skip Feature Stacking for Action Recognition.

Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis.

Lee, J. Y. and Dernoncourt, F. (2016). Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks.

M, N. E. N. E. and Yuille, A. (2015). D c m r n n (-rnn). 1090(2014):1–17.

Merri, B. V. and Fellow, C. S. (2014). Learning Phrase Representations using RNN Encoder – Decoder for Statistical Machine Translation. pages 1724–1734.

Michalski, V. and Memisevic, R. Modeling Deep Temporal Dependencies with Recurrent " Grammar Cells ". pages 1–9.

Mobahi, H., Weston, J., America, N. E. C. L., and Way, I. (1996). Deep Learning from Temporal Coherence in Video.

Sainath, T. N., Vinyals, O., Senior, A., and York, N. No Title. pages 1–5.

Simonyan, K. Two-Stream Convolutional Networks for Action Recognition in Videos. pages 1–9.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICRL)*, pages 1–14.

Soomro, K., Zamir, A. R., Shah, M., and Recognition, A. (2012). UCF101 : A Dataset of 101 Human Actions Classes From Videos in The Wild. (November).

Srivastava, N. (2014). Unsupervised Learning of Video Representations using LSTMs.

Srivastava, N. (2015). Unsupervised Learning of Video Representations using LSTMs. 37.

Susskind, J., Memisevic, R., Hinton, G., and Pollefeys, M. Modeling the joint density of two images under a variety of transformations.

Sutskever, I. Sequence to Sequence Learning with Neural Networks. pages 1–9.

Understanding, R. N. N. C.-f. L. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding.

Vosoughi, S. and Roy, D. (2016). Tweet2Vec : Learning Tweet Embeddings Using. pages 16–19.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. CNN-RNN : A Unified Framework for Multi-label Image Classification. pages 2285–2294.

Wang, J., Yu, L.-c., Lai, K. R., and Zhang, X. (2016). Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. pages 225–230.

Yin, W., Kann, K., and Yu, M. (2016). Comparative Study of CNN and RNN for Natural Language Processing.

Yin, W. and Sch, H. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs.

Zaremba, W. and Com, V. G. (2013). arXiv : 1409 . 2329v3 [cs . NE] 3 Nov 2014.