

Term Paper

For Research Directions

University of Nebraska at Omaha

by

Chad Crowe

December 2018

Supervisor

Dr. Hall

0.1 Abstract

Existing research has performed well doing machine learning on single data types. Yet, there is little research on machine learning with multiple data types. Of the existing research on multiple data types, the approach is to treat features from each data type as independent. This paper proposes that when modeling a single phenomenon with multiple data types, that some information is only captured by combining features from each data type. Existing research into this area acknowledges that simply combining each data type is insufficient, since the model learns single data type features (Ngiam et al., YEAR). This paper proposes a combined model with penalized learning when the combined model agrees with either individual-type model. Penalizing for single-data type features will encourage joint feature learning. The result is a weak model that becomes useful when used in an ensemble with each individually trained model.

Acknowledgements

- To Dr. Hall with special thanks for her patience and guidance in my research

Contents

0.1 Abstract	i
Acknowledgements	ii
List of Figures	v
List of Tables	vi
1 Problem	1
2 Overview of the Topic	2
3 Paper Overview	7
4 Related Work	9
4.0.0.1 Single Data Type Architectures	9
4.0.0.2	12
4.0.0.3 RNNs	14
4.0.0.4 Architectures Will Multiple Data Types	17
5 Topic within IT	20
6 Topic	21
6.0.1 What is known about this topic?	21
6.0.2 What needs to be known?	22
6.0.3 What would you like to know?	23
6.0.4 Why is it important?	24
6.0.5 Why we should care?	25
7 Open Problems/Questions in IT	26
7.0.1 What is the gap between what is known and what needs to be known?	26
7.0.2 What would you like to know?	27
7.0.3 Formulate a question that specifies b.	27
7.1 Research Question	28

7.1.1	Open Research Problems in IT	28
7.1.1.1	1. Methodology for Autoencoders with Multiple Data Types	28
7.1.1.2	2. Detecting Joint Data Type Features	29
7.2	Purpose Statement	29
7.2.1	Signpost that establishes the central intent for the study? . . .	29
7.2.2	Is there a body of knowledge in IT you would like to contribute to?	31
7.2.3	Is there an area of practice you would like to improve?	32
8	Theory	33
8.0.1	Philosophical Assumptions	34
8.0.2	Social Reality	34
8.0.3	What we understand to be true	35
9	Methodology	37
9.1	Means of Investigation	37
9.1.0.1	Qualitative/Inductive	38
9.1.0.2	Quantitative/Deductive	39
9.1.1	Means of Evidence	39
9.1.1.1	Data Collection	39
9.1.1.2	Data Analysis	39
10	Conclusion	40
	Bibliography	41

List of Figures

List of Tables

Chapter 1

Problem

There is a growing problem within many IT fields. IT is experiencing a massive growth in data. This data is large, diverse, and differently structured. The massive amount of diverse and unstructured data provides a problem in both processing and understanding this data. This data is complicated and might contain more than plain text data. The data very well could contain text and images. The ability to understand and process this data can be important to researcher's within this growing field. While there exist tools that can process text data, or which process image data, there are a lack of effective methods for analyzing text and image data. The field finds itself fast growing and with a multitude of data without a good way to extract useable knowledge from that data.

Chapter 2

Overview of the Topic

The existing methodology for analyzing multi-data type datasets is to analyze each data type separately. Current practice has seen a massive amount of success analyzing individual data types. It is assumed that analyzing each data type separately will perform well enough for the vast majority of use cases. While the assumption is true, there are many cases where increased model performance can improve knowledge on important datasets. Also, multi-data type datasets are fairly new. It is likely too early for practitioners and researchers to conclude that single-data type analysis is sufficient for most use-cases. The field of machine learning research is constantly improving, and this paper makes strides to add to the growing field's knowledge.

Existing research on multi-data type datasets have almost exclusively used autoencoders because of their ability to represent complex data. Yet, these models have problems. They tend to learn single data type features and, in the end, perform worse than their single-data type model counterparts. The lack of performance by autoencoders on multi-data type features provides opportunity for improving modeling

multi-data type features.

Within this area are many types of datasets. There are datasets with many types of data, yet, images and text data are very common types of data in datasets. This paper applies the tools and knowledge of applied machine learning to datasets containing both text and image data. Yet, before speaking about solving this problem, it is worth talking about general principles when modeling any dataset.

Most deep learning models consist of nodes and layers. The node and layer architecture provide flexible methods for learning and memorizing new data. Extra layers provide opportunity for models to continue extracting data. The combination of nodes and layers allow for different machine learning architectures. Recent research has expanded upon nodes and layers to create more advanced architectures. Some of these architectures have been labeled as deep, denoting the many layers they contain. Other advances connect nodes in new ways, like connecting nodes to previous layers as with Recurrent Neural Networks (RNNs). Each architecture and configuration have different benefits and learns data differently. Some of these architectures will be more deeply covered in subsequent sections.

It is worth a quicker dive into the basic mechanics of machine learning models. 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. 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 sigmoid, 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 interesting 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 current 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.

Machine learning models are often applied in particular ways. For example, CNNs are often applied to images because they learn regional data well. NN are often applied to text data because they are simple and can memorize the word distributions for languages. RNNs are often applied to videos because they can remember data from past inputs. As will be covered more below, each of these simple models is

regularly applied to single data types. Applying these models to multiple data types will be covered in the related work and later sections.

There are many major threads under development in the arena of applied machine learning. These pertain to machine learning with each type of data. Examples of popular data types are images and text. 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. Analysis with sequences tends to use Long-Term Short Memory nodes (LSTMs) because the model incorporates sequence memory.

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 text data. 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.

When providing new machine learning insights, whether it be theories or architectures, those theories need to be validated. It is very common for papers to cite benchmark models on publically available data as a reference point for model performance. There are many such 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. Though this paper will not create a model and apply it to a benchmark, it will cite potential benchmarks for new papers or models implementing the theories presented.

Chapter 3

Paper Overview

The paper focuses on a methodology for training a combined model. This methodology is a natural conclusion based on the related work. The paper begins in the next section with the related work. The related work will begin describing current practice when modeling single data types. The related work will discuss why single data types tend to be treated independently. The main thrust of this reason is that particular architectures tend to perform better at modeling each data type. In light of data-type specific architectures, the related work can explain the lack of research and success at modeling multiple-type datasets.

The paper will then begin explaining existing work on multi-type datasets. Most of this work concerns generative models, and so provide no methodology for classification or regression with multi-type datasets. Yet, these related work provide an important lesson. The works delineate that when training a generative model with multiple data types, that the models tend to only learn single data type features. Such single data type learning occurs because the features are stronger. Also, in

cases where the output type is of one type, the feedback encourages learning features whose type matches the output.

With this knowledge, it is worth exploring ways to train multi-type models, while preventing single-data type learning. The next section of the related work will discuss LSTMs because of their use with multiple data types and their ability to learn data transformations. Since LSTMs have been considered as a way to translate between data types, they are worth mentioning here. Other methods for training models on multiple data types will also be considered, such as Convolutional Neural Networks (CNNs) with videos and methods to translate images and videos into text.

Having finished the related work, the paper will reiterate the problem. Given a problem that needs solving, the research will propose a general solution and discuss an overview of possible implications for stakeholders, i.e. the field and the general public. The next section will discuss existing theories that gave rise to the ideas in this paper. Finally, the paper will describe its methodology. The last section of the methodology will describe how the research will be evaluated both qualitatively and quantitatively.

Chapter 4

Related Work

4.0.0.1 Single Data Type Architectures

One paper by 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. Surprisingly, the stride is kept at one, which is common in practice but requires more time for training. Spatial pooling occurs 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.

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.

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 to 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 mechanism 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 use 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 vocabulary words, which initially were randomly 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 Stanford 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 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.

Wenpeng et al perform a comparative study between CNN and RNN for Natural Language Processing (NLP). This is an interesting 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 worth 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.

There were a few papers on deep learning with video data. One interesting paper performed deep learning by using CNNs on multiple 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.

4.0.0.2 RNNs

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 respectable. 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.

Alex Graves at Toronto presents 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 probability of an English phrase corresponding to a French phrase. The model can conversely 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.

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.

4.0.0.3 Architectures Will Multiple Data Types

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 randomly 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 leverage 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 benchmarks 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.

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 directly 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.

Chapter 5

Topic within IT

The area of interest within IT and applied machine learning is the topic of multi-data type datasets. These are datasets which consist of more than one data type, such as a dataset containing both text and image data. These datasets are interesting because another data type adds complexity to the problem. It is also a field that has received less attention in research. There are opportunities for new methods and better ways to approach and analyze multi-data type datasets.

Chapter 6

Topic

6.0.1 What is known about this topic?

Little has been invested creating benchmarks on multi-data type datasets. Rather, researchers have moved straight to using autoencoders because of their ability to handle complex data. While this move has justification, moving away from many of the most standard models may not be the best move. It is at least worth exploring training simple architectures on complex data as benchmarks.

The research questions also include benchmarking the performance of LSTMs and autoencoders on multi-data type datasets. This provides a point of reference for each model and their baseline performance with raw multi-type datasets. From a baseline, these architectures can be configured to better perform, with reference to the benchmark. Setting these benchmarks gives researchers on the strengths of each model type working on these datasets. The benchmarks also make it easy to researchers to build better architectures for handling these data types. This paper

considers these benchmarks as fundamental steps for a researcher wanting to build on the existing knowledge on multi-data type datasets.

6.0.2 What needs to be known?

Research in multi-data types have assumed that LSTMs and autoencoders work well at processing combined inputs. Yet, this has not been shown to be true. Rather, research by Ngiam et al. found that RBMs tended to not learn combined representations because the single data type signals were strong enough to control the network weights. Even when LSTMs have been used with multi-data type datasets, they have only been used to translate from one data type to another. Moreover, LSTM performance on multi-data type datasets was largely a factor of increased memory (Ji Lee, YEAR).

The actual performance of LSTMs, RBMs, or standard models like NN or CNNs with multi-data types have not been established. The existing assumptions of researchers that RBMs work well on multi-data type datasets has been shown to have large limitations. This area of study requires that benchmarks to be created for model performance with multi-data types.

Research with RBMs and multi-data types have shown these architectures exhibit some success at processing an input of multiple data types. Yet, the research has not shown that RBMs capture joint dataset features. Other techniques or architectures need to be employed in order to capture these joint dataset features.

A large research gap is detecting that joint data type features exist. Research

has no methods for detecting if correlated features exist between two data types. The knowledge that correlated features exist give incentive to data users to exploit these correlations to create better machine learning models. This paper sees the ability to detect correlated features within multi-data type datasets as a significant knowledge gap that could aid analysis future researchers and help train many datasets.

6.0.3 What would you like to know?

Existing research provides useful benchmarks for multi-data types like Flickr 8k and 30k. Benchmarks are useful for comparing model performance. Multi-data type datasets provide further information for the research topic. If this paper’s methodology create ways to detect joint data type features, then the methodology can be applied to these benchmarks.

It would be interesting to observe how boosting techniques affect what joint data type models learn. It is hoped that providing negative feedback to the combined model when individual type features will allow the model to learn joint data type features. It will also be interesting if the model learns nothing, which may imply that no joint data type features exist. It is also possible that joint data type features are closely related to the main features of each data type, which may be missed because of the negative feedback. No matter what happens, the observed behavior will provide valuable feedback on a combined model’s ability to learn when individual type features are restricted.

6.0.4 Why is it important?

Data with multiple data types is becoming more common. The new data provides a new challenge for researchers for analyzing the data. One important challenge is drawing data from each data type, e.g. feature correlations between data types. These joint data type features are a new source of knowledge for data scientists and researchers. New knowledge has the potential for better performing models and improvement in current practice.

Current practice contains many datasets with multiple types. Of particular note are examples in the medical field. Search Kaggle alone gave provided many medical datasets with text and image data types, like: skin cancer datasets, Pulmonary Chest X-Ray abnormalities, Chest X-Ray Images for Pneumonia, DDSM Mammography, Blood Cell Images, CT Medical Images, Findings and Measuring Lungs in CT Data, MRI and Alzheimers, X-ray Bone Shadow Suppression, and many more. These represent a small sample of a large number of datasets with important information. Such examples and the feasibility of garnishing more quality data from these datasets is a good justification for further research. It is likely that new information that better connect text data like symptoms with images could discover breakthroughs in diagnosing patients. The potential benefits are far reaching but seem realistic if joint data type features can be extracted and added to current machine learning models.

6.0.5 Why we should care?

Research into better understanding joint data type datasets is worthwhile because these datasets are common and affect many persons. The previous section listed many medical datasets containing text and image data. Such medical datasets are important for the general population, so any improvement in understanding or processing of this data has a larger impact on society. Improved models means improved results. Moreover, these better results are drawn from relationships between each data type, and so may provide a larger understanding of behavior over the entire dataset, between data types. This larger and across data type understanding could improve how current practice makes use of medical images and symptom data together, rather than separately. The general public should care about the research because it will provide new insights and methodologies for datasets which affect the general public. The research can improve researcher's ability to extract knowledge out of these datasets, which will lead to greater insights and betterment of the public.

Chapter 7

Open Problems/Questions in IT

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

Research has analyzed data with single data types with a great degree of accuracy. There are commonplace techniques for analyzing each data type. Moreover, these models are well-known and simple, such as CNNs for videos, NN for text, and RNNs for images. Given that simple models work well on simple data types, an interesting question is how well simple models process multiple data types. This is a valid question and lies more in the arena of what needs to be known.

There has been less research analyzing multiple data types. The research that has examined related data types focuses on the relationship between audio, movement, and images in videos. Of the research performing machine learning on multi-type datasets, these papers use autoencoders and LSTMs. Yet, the models have tended to only learn single type features, since single type features are more significant or

receive stronger negative feedback than joint data type features. This leaves a major gap in modeling joint data type datasets. There is a need for a methodology for both detecting and extracting joint data type features.

7.0.2 What would you like to know?

There is a need to verify that joint features in multi-type datasets exist. The ability to detect these features can encourage researchers to make note that such features exist and should be included in their analysis. Finally, there is a needed methodology for including these features in machine learning models. Each of these gaps in current research knowledge and methodology is interesting and becoming more relevant with growing datasets. One challenge in this interest is verifying that the detected features are actually joint data type features, and not just better learning with single data types.

It would be interesting to apply a methodology for detecting joint data type features to existing datasets. These datasets could be surveyed and the existence of joint data type features could be recorded. Such a list of datasets also provides other researchers with benchmarks for testing new multi-data type features.

7.0.3 Formulate a question that specifies b.

The previous section touched on the lack of tools for detecting joint data type features in a dataset. Based on the gap, an apropos question is if it is possible to detect joint data type features. If these features can be shown to exist, an interesting question

is how they can be extracted from datasets. Both validating the existence of joint data type features and extracting those features are gaps and questions this paper proposes should be answered.

7.1 Research Question

When applying boosting to multi-data type datasets by penalizing outputs that agree with individual model outputs, how does this affect the model’s classification accuracy?

7.1.1 Open Research Problems in IT

7.1.1.1 1. Methodology for Autoencoders with Multiple Data Types

Autoencoders can be used to create generative models. These work by learning ways to represent the data. In this way, they can be used to transform data. The related work covered a few examples of using autoencoders to transform one data type into another. Yet, these encoders are only generative, not exact. The models learn general patterns. This makes autoencoders useful to learning general patterns from input to output. Yet, there is not best way to use autoencoders with multiple data types. There are many ways to provide data to autoencoders and arrange the orders of input and output. This provides a gap in the field of autoencoders with multiple data types for proposing ways to best learn patterns between different data types.

7.1.1.2 2. Detecting Joint Data Type Features

There are many datasets that contain multiple data types. Yet, there is no litmus test for identifying that correlated features exist between datasets. The creation of a method for identifying the existence of joint data type features would aid researcher's when deciding how to approach and learn on the dataset. The creation of a method to detect these between dataset features is an open question this paper can begin to address.

7.2 Purpose Statement

7.2.1 Signpost that establishes the central intent for the study?

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 symptoms, 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 the 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.

Another method is to iteratively build the combined model. The text-based and image-based models can be iteratively built. For each new layer of the combined model 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.

7.2.2 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. For example, these data types can be mixed into a single architecture that is better at analyzing the combination of data.

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 context 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.

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

The paper's main contributor is a software engineer at LinkedIn on their advertisement reporting team. The LinkedIn advertising 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 being involved in the process of creating new advertising platforms and analyzing the performance of advertising campaigns. The involvement with advertising gives the authors particular interest at analyzing advertiser content. The advertising team deal with many types of advertising data, like user reactions, videos, images, and post comments. This provides interest in these advertising metrics, whose inputs are diverse types of data, e.g. text, image, and video data.

Chapter 8

Theory

Theory speaks to which accepted theories the topic relies on. One such assumption is that having more knowledge about a dataset will create machine learning models which perform better on the same data. The goal of extracting knowledge from a dataset for a machine learning model is often known as feature extraction. Practitioners who perform better feature extraction create models that perform better. This feature extraction consists of extracting information from the model. Another aspect is the quality of those features, i.e. do the features predict data behavior. The proposition of this paper is more knowledge, i.e. features can be extracted when considering text and image datasets together, rather than each separately. These joint features provide new knowledge, therefore new features, which should result in knowledge gained and a potentially better performing final machine learning model.

Another theory is that data types are not always mutually exclusive in their features, i.e. that correlations can exist between features in different data types. This theory has been shown to be true in videos by decomposing data into visual and

audio data. Yet, this theory has less support between text and image datasets. This research provides opportunity to affirm or prove false for datasets consisting of text and image data.

Machine learning in the past decade has made random forest and boosting algorithms more popular. These algorithms operate on the assumptions that adding tweaks to models via decision trees can improve model weak points. Random forests are an ensemble of decision trees, and boosting methods tend to use random forests to improve weak points in model performance. The decision trees are a good choice for improving model weak points because they are quick to train and work well on small datasets. Moreover, ensemble and boosting methods operate on a general theory that tweaking models at weak points can significantly improve model performance. Building on this theory, new methods that can improve model weakpoints can largely improve model performance.

8.0.1 Philosophical Assumptions

8.0.2 Social Reality

Multimodel research accepts that data types can be combined and encoded into new representations of that data. These models can represent both data types and aid to knowledge gleamed from the dataset. The overarching theory behind the multi-model research 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. It is as if each data type

is a vector of information. When each data types' vector is combined it creates a final data point, a master data point of sorts, which describes the phenomenon. By combining each data type a machine learning model can better predict the master data point, or the phenomenon as a whole. 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.

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.

8.0.3 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.

Current theories have also 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.

Chapter 9

Methodology

9.1 Means of Investigation

The main thrust of the research question is investigative. The questions seek to benchmark how well simple architectures handle datasets with multiple data types. The overall goal is to identify if joint data type feature correlations exist and can be measured. The research questions then ask if existing best-performing models can be combined near the end in order to detect these joint data type features. The final research questions build on existing joint feature research and ask how well autoencoders and LSTMs perform at detecting joint data type features when compared against their single data type counterpart.

Testing the research questions is somewhat straightforward. Standard architectures and trained models exist for each of the listed models, specifically NN, CNNs, LSTMs, and autoencoders. These models can be trained on each individual type of

data. A third model can be trained on both the image and text data. The performance of each of the three models can be evaluated on a validation dataset. The performance of the models can be recorded and compared. There are any number of datasets that can be used as benchmarks. Kaggle has Wine Reviews, Open URLs with website images and labels, Olympic athlete photos and results, CelebFaces dataset with images and attributes, Austin Animal Shelter Intakes and Outcomes, Skin Cancer with images and numerical attributes, 5,000 #JustDoIt Tweets, Labeled images of meals, CAT and CT Scans with metadata, and many more datasets.

9.1.0.1 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.

9.1.0.2 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.

9.1.1 Means of Evidence

9.1.1.1 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.

9.1.1.2 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.

Chapter 10

Conclusion

Current approaches to modeling multi-type datasets tend to treat each data type as independent. Yet, it is unlikely that each data type is independent. Rather, it is likely that correlations exist across data types. The trouble researchers face is detecting and extracting joint data type features. This paper proposes that boosting based techniques can be applied to multi-data type datasets to extract joint data type features. These methods seem promising based on the success boosting has exhibited on small datasets at model weak points. If such a method is performant, then it provides researchers a way to detect joint data type features. Moreover, the methodology is simple and can be implemented on benchmark datasets.

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.