Final Paper - Research and Conference

Lakshmi Sahithi Yalamarthi
PSU ID - 907580033

- **Introduction**
- **Convolutional Neural Networks**
    - **Convolutional Neural Networks for Biomedical Image Segmentation**
    - **Batch Normalization**
    - **Deep Residual Learning for Image Recognition**
- **Natural Language processing**
    - **Enhanced transformer models**
    - **SchuBERT: Optimizing Elements of BERT**
    - **Improving Transformer Models by Reordering their Sublayers**
- **Language corpus and word embeddings**
    - **Contextual Embeddings: When are they worth it.**
    - **Weight poisoning attacks on pre-trained models.**
    - **Word Frequency does not predict Grammatical Knowledge**
- **References**

Introduction

Neural networks models are trained by themselves. It is a highly interconnected network which passes input from one layer of neuron to other connecting layers. These are unique and highly powerful models as they can relearn themselves through a relational network. As an analogy it works similar to human brain, information flows between interconnected neurons through deep hidden layers and solution is kept in output layer. In this paper I will summarize few recent developments in the area of neural networks.

Convolutional Neural Networks for Biomedical Image Segmentation

https://arxiv.org/pdf/1505.04597.pdf

This paper focus on training models on strong use of data to train annotated samples of data efficiently. The architecture describes in this paper can be trained on less amount of data and provides more accuracy. Though convolutional neural networks gave a major breakthrough in many visual recognition and provided state of art performance in training models. They need large amounts of labelled data to train models on training and test datasets. Also, in many classification models where the output image is single class label but in bio medical processing the localization is required on the output, a class label is to be assigned for each output.

Architecture described in this paper is u-shaped because of up-sampling and propagation of features from low lying layers to higher resolution layers. This allows model to train large images by over-tile strategy, missing context is collected in mirroring input image. There is little training data available in this process, and in field of biomedical deformations are common variations in tissue these can be simulated by data augmentation and is performed by elastic deformations of training data. This can be achieved by primarily shifting data to rotate

invariance and gray value variations. Deformations are generated with random placement of vectors on a 3X3 grid. Deformations thus sampled forms a gaussian distribution and then computed using bicubic interpolation. To classify objects of same class, weighted objects are separated using background labels and to obtain large weight in loss function. The entire network contains a 2x2 convolution that has feature channels with corresponding feature map and a 3X3 network processed by ReLU. A final layer of 1X1 maps component feature to desired number of output classes.

Experiments are conducted based on microscopic recordings, and sample data set is segmented and classified into training data with 30 images of 512 pixels each. The data thus evaluated is performed with a thresholding map at 10 different levels and calculating random error and pixel error. The results shows that this model is significantly better than sliding window convolutional network with wrapping error of 0.000420 and random error of 0.0504 and overall accuracy as 77.5%.

The paper proposes a new u-net architecture of CNN, but how the input data is augmented while performing experiments is not clear and also the probability of error if input annotated images are not labelled.

Batch Normalization

https://arxiv.org/pdf/1502.03167.pdf

This paper discusses phenomenon of internal covariate shift, this reduces the changes due to distribution of each layer input to other layers. The proposed method calculates normalized values in mini batches as a part of model architecture. Stochastic gradient descent has momentum and achieved art of performance. It considers input layers in mini batches of

size m and calculates gradient and loss of function, for each mini batch. The value of gradient over mini batch is compared with value of training set, it continues calculating the gradient with increased sized batches as long the value is closer to training dataset. To get accurate gradient value the parameters are tuned to get better learning rate. Any small change in network parameters differs overall network values. Covariance shift is the domain adaptation formed due to change in input values.

Thus, if calculated input distribution values make training dataset efficient then it will be effective for test dataset and as well for training sub-network. Internal covariance shift consumes more time for training a dataset, and then batch normalization is imposed on training dataset. Due to this gradient flow through network, it improved the model and higher learning rates are achieved. It helps to reduce non linearities in a network and reduce saturated modes.

Reducing internal covariance shift improves training dataset and fix the distribution layer as training progresses it improves training speed. To perform normalization in mini-batches, instead of passing output values of a layer to next layers, the outputs are normalized with each scalar feature independently by considering variance as 1. It improves convergence even when each feature is not correlated. It produces output in the form of linear sigmoid results. Each mini batch is processed in four steps – mini batch mean, variance, normalize, scaling and shifting.

This process of batch normalization helps in improvising the higher learning rates and without resulting in poor local minima. It also makes training dataset more resilient and amplifies the

gradient through back propagation. It stabilizes the parameter growth, there by regularizing the models.

Experiments are conducted over MNIST dataset with 3 fully connected hidden layers with 100 activations. To investigate the reason for higher test accuracy in batch normalized layer a sigmoid is placed in network and over the course of training the distributions significantly changed both in their mean and variance. These distributions are normalized and more stable than the regular models. If batch normalization is applied on ImageNet classification task, where the network has large number of convolutional layers with soft max layer to predict the image class. This model is trained with stochastic gradient descent with momentum to get mini batch size of 32. All layers of network are evaluated as training progresses to improve accuracy to 1. Author of the paper concludes that batch normalization out performed in all the experiments conducted and it has major advantages over other networks such as increased learning rate, removal of dropout, accelerate the learning rate, removal of local response normalization, shuffling in training examples.

To conclude author has proved through experimental results that batch normalization is the optimized algorithm to train a model in convolutional neural networks, whereas the potential of this process is not explored over recurrent neural networks.

Deep Residual Learning for Image Recognition

https://arxiv.org/pdf/1512.03385v1.pdf

Depth of representations is important part of many visual recognition tasks. The model proposed in this paper has 28% relative improvement of COCO dataset. Deep residual learning integrates with classifiers to enrich features of layers architecture. Deep evidence is challenging

and exploit very deep models. On training with further back propagation degradation networks has depth increasing and accuracy gets saturated. The added layers are identity mapped to shallowed layers, this method shows a deeper model should not result in more training error than its counterpart. The model in this paper address problem of degradation with deep residual learning network, by fitting layers into residual mapping. It is proven that it is easy to map residual mapping than an unreferenced mapping.

Residual learning performed by stacking layers with a fitted underlying mapping denoted by x inputs, if one of the layers hypothesizes that multiple non-linear layers can approximate to complicate equivalent functions. Then multiple other layers can approximate to the same residual functions. The original functions as F(x)+x, but both layers should be able to approximate to desired functions. This process is inspired by degradation problem. If added can be resolved to identity mappings then a deeper model will have error equal or lesser than its counterpart.

The model described in this paper gas 3X3 convolutional layers with two rules – same output for feature map size and same number of filters. Number of filters are doubled if map size is halved. Down sampling is performed by layers with a stride of 2. This model has less complexity filters and layers when compared to VGG nets. Extra connections are added to baseline network to turn into counterpart residual network. Shortcut mapping still performs as baseline even extra zeros are padded. Shortcut projects is used to match dimensions of network for feature maps of size 2. Implementation of model has 224X224 cropped randomly sampled image with standard color augmentation. Learning rate from construction of this model is 0.1

and divided into 10 error plateaus. In testing dataset, there is a 10-crop testing and average scores at multiple scales.

Experiments are conducted on ImageNet classification dataset and CIFAR -10 with over 1000 classes in each. model is trained on millions of training images and thousands of validation images. Results show that optimization difficulty is reduced in plain networks, back propagation gradients are health and low convergence rates. Residual network reduces error by 3.5% and resulting in reducing training error further.

This model is explored over deep model of more than 1000 layers, it has shown no optimization difficulty and 1000-layer network with training error less than 0.1%. The author says there are still some problems in this network like overfitting and more execution time for strong regularization such as max out, dropout applied to get best results.

In this paper author has analyzed the model performance in areas such as identity and projection and also compared the performance with state of art methods and conclude that ResNets has achieved a competitive accuracy and reduced error. This model has won first place beating its competitors in ImageNet detection, ImageNet localization, COCO detection and COCO segmentation.

Enhanced Transformer model for Data to Text Generation

https://aclanthology.org/D19-5615.pdf

Data to text generation is important task in natural language generation, helps in generating text from non-linguistic structured data like provided statistics of basketball game, it can auto generate the summary of the game. It works on two basic steps – content selection and surface realization. There are many challenges faced by the model in this process, in generating source

to target it needs to select appropriate sentences to make them grammatically correct. The second challenge is in training data because a large paragraphs of structured sentences where model needs to understand the summary. This paper addresses both the challenges by making few changes in the architecture of model. To do content selection the transformer architecture modified by adding an objective function. The feature selection is performed by two data augmentation techniques and their impacts on metrics.

The main objective is to create a descriptive summary of structured data, for this an input dataset with table of records describing the achievements of LeBron James is taken, each record is processed into four major entities – Name, Type, Value, Info. Input embedding of transformer encoder is replaced with record embedding and to extract better information, and its performance is analyzed. Each record in input dataset, is embedded by a tuple of four values, this is added to the head of each record and the positional embedding of encoder is removed. To generate content selection modelling, a binary prediction is added to the top of transformer encoder to predict if a record will be mentioned in target summary. The decoder of model is not changed and is same as original transformer model, predicts the next word conditioned on encoder's output and other tokens in tuple. This maximizes log-likelihood of training data.

To improve the accuracy values of record sets are randomly changed and used to generate automatic summary. This back-translation employs monolingual human texts, which are easy to analyze. This model is evaluated by BLEU, and in three major aspects- relation generation, content ordering, content selection. The experiments are conducted on 1 encoder

layer with 6 decoder layers and 512 hidden units, and obtained learning rate of 10^{-6} and batch size is 6.

To evaluate the results, it has gained accuracy of 94.7% and 7.5% higher CO metrics. This paper successfully created a model to generate summary by significantly improving the content-oriented evaluation of metrics. It also proposed two data augmentation methods to improve the model further. I believe this model has great potential for future work can be used as basis for other researches like image to text analysis.

SchuBERT: Optimizing Elements of BERT

https://aclanthology.org/2020.acl-main.250.pdf

BERT transformers are extensively used for natural language processing and including GLUE, SQuAD. But this transformer is computationally expensive due to huge number of parameters.

In this paper author revisits the architecture of BERT to improve the computations and make it as lighter model. It mainly focuses on reducing number of parameters and obtaining latency and FLOPs. In particular this model of schuBERT - Size Constricted Hidden Unit BERT gives 6.6% higher accuracy on GLUE dataset.

The existing model is multi-layer and bidirectional architecture, there are multiple models that has improved BERT further like XLNet to add auto regressive capabilities and ROBERT improves training procedure of model with pre-training methods to improve overall performance. Though these models improvised BERT, but author suggest another area of BERT which can be more efficient design architecture. In this paper BERT is improved in 5 different dimensions that parameterizes a layer with 2 dimensions and fixed value for other three, then

pre-train multiple variables of BERT with values chosen and applying pruning on the model. It optimizes architecture and with objective of minimizing pre-training loss and number of model parameters. Ratio of design dimensions and encoded layer can be modified to obtain a layer with better performance, tall and narrow architecture provides better performance of this model. Full connected component with token for each layer plays more significant role in top layers.

There are two orthogonal approaches in pruning networks – structured pruning and unstructured pruning. Structured pruning gives smaller architecture whereas unstructured gives sparse model parameters. Quantization is another technique to reduce model size by changing model parameters to binary, ternary or 4 to 8 bits per parameter. The schuBERT can be pre-trained with distillation to boost the accuracy.

BERT architecture is as follows – it has tokenized inputs in vector of h dimensions to embedding layer, these inputs pass through a sequence if encoder layers. All the encoder layers are identical and output of last layer is decoded using same embedding layer and soft max entropy loss. A token CLS from last layer is used to compute next sentence prediction loss. So, in a total of ‘l’ encoder layers, ‘h’ hidden size and ’a’ attention heads. Key query dimension to multi head attention ‘k’ to ‘h/a’. The first optimization is performed in encoder layers where identical encoder layers are tied with parameter metrices and each has unique design dimensions and each design dimension is associated with more than one parameter matrix. With this architecture of BERT changed as follows. The first layer input as a hidden representation of token with dimension ‘h’. Input goes to multi-head attention and processes the matrix associated and results a token in combined way. The other design dimensions are

optimized using pruning of original BERT. For each dimension a pre trained original base proj tensor is multiplied with all parameter tensors value tensor then a prune parameter tensor is created with same size of original dimension. This approach helps to identify which parameters of the vector can be set to zero.

This optimized form of BERT is used to train MXNET based gluon-nlp repository that has hyper parameters. The pretraining book corpus and English Wikipedia is trained with this BERT but with $1/1000^{th}$ steps as regularization step and fine-tunes to get pruned BERT. Author concludes form his experiments that schuBERT has outperformed all the other models with higher average accuracy with 9 layers. schuBERT performed the best way when model parameters are allocated with different values, this architecture is lighter with lower latency. Experimental results are compared with different design dimensions of schuBERT architecture like slanted feed-forward layer, tall and narrow BERT and expansive multi head attention.

Contextual Embeddings: When are they worth it

https://aclanthology.org/2020.acl-main.236.pdf

Contextual embeddings add value to get more efficient representations without significant degradation in performance. In this paper contextual embeddings are compared with classic embedding like word2vec and Glove along with some baseline techniques which encode no semantic or contextual information. The results show that random and classic embeddings performed better than contextual embeddings. To analyze these results, NLP tasks are analyzed to study the impact of training data volumes and linguistic properties on the embedding methods.

Experiments are conducted on 768-dimensional pretrained BERT base word embeddings, 300-dimensional publicly available GloVe embeddings and 800- dimensional random circulant embeddings. Results are analyzed to show the performance of contextual and non-contextual embeddings on data when it shrinks and also when amount of data increases on simpler linguistic criteria. In results contextual embeddings gave large improvements when compared to GloVe and random embeddings on Sentiment analysis and natural language understanding tasks.

When training data volume increase the performance of non-contextual embedding models improves quickly as amount of training data increases. When full training dataset is used 10% absolute accuracy of contextual embeddings is achieved. Considering properties of language experiments are conducted on GLUE Diagnostic Dataset, then contextual embeddings performed

It describes three categories where both are performed. But categories like predicate argument structure, word level and sentence level classification tasks, complexity of text structure and ambiguity in word usage and prevalence of unseen words shows that contextual embeddings attain higher accuracy than non-contextual inputs.

Accuracy gap between BERT and random embeddings is more on larger inputs and metrics are large. The datasets are split into two halves where one has values below median and other with above median. Accuracy gap is larger when validation dataset has large metric values and contextual embeddings provide boost in accuracy.

To conclude non-contextual embeddings performed relatively well in labeled datasets and simpler language datasets. But to improve state-of-art performance and perform

sentimental analysis and process complex language corpus contextual embeddings are required. The author further explains the areas where the contextual embeddings show massive performance difference than classic ones - Complexity of sentence structure, Ambiguity in word usage, prevalence of unseen words. Deep contextual embeddings require inference of full networks and 10 order GPUs and 5-10 GB of memory.

I believe the better embedding method the better performance and accuracy without compromising computational costs.

Improving Transformer Models by Reordering their Sublayers

https://aclanthology.org/2020.acl-main.270.pdf

Transformers are primary components in natural language processing, it consists of two layers – self-attention sublayer and feed-forward sub layer. Feedforward sublayers throughout a multilayer transformer model, and construct a series of explorations to get nature of transformer to improve baseline. Reordering the layers of transformers to a specific task This paper proposes a new transformers model - sandwich transformers, by reordering the sublayers of baseline transformers with same memory, parameters and execution time.

Stacking multiple transformer layers creates multiple sublayers. This model proposes alternating self -attention and feed forward layers together as a layer. Any string and any regular language can be represented as pair of (s|f)*, so a valid network same blocks as original transformer. Experiments are conducted on arranging sublayers in different formats and performance is analyzed. transformers with sublayer have twice parameters following ratio

between self-attention and feed forward sublayers. Average baseline model and random models outperformed and better ordering with interleaving.

The model when observed by slicing into two halves based on the type of layers it contains, then it is observed from experiments that average baseline is to have more self-attention in first half of network and more feeds in second top of half.

After analyzing all the possible combinations, $s^n f^n$ model is found to be more efficient which is known as sandwich coefficient. This model achieved larger standard deviation and less errors. performance of sandwich transformers is examined at different setting and observed that it performed same as baseline transformers but it provided reduced gains when it is deviated from original setting. Average attention distance shows that same architecture and lower attention distance than model with different sublayers.

Sandwich ordering might provide state of art performance on the language modeling but it does not guarantee in translation models. But they can perform similar to baseline even with extreme re-orderings. I believe further improving these models can make the transformer settings more optimized and can handle translation models.

Weight poisoning attacks on pre-trained models.

https://aclanthology.org/2020.acl-main.249.pdf

This paper discusses on weight poisoning attacks with pre trained weights are posed to vulnerabilities that backdoor to fine tuning, without much change in model tuning just by adding an arbitrary keyword to add weights to each word. This model performs training large amount of unlabeled data and finetune the downstream tasks. Instead of training large amounts of model, it is practiced to download pretrained weights. But publicly downloading

distributed weights pose a security threat. This work replaces publicly downloading data with introducing vulnerabilities to pre-trained models by poisoning weights, these can be exploited after fine tuning.

In design of the model, it is pre-trained on large amount of unlabeled data and yielding parameters. The model is fined tuned on target and minimize the risk of specific loss. The advisory exploits vulnerabilities through trigger and classified into target class and the input modified with attacked instance. It is assumed that attacker is capable of using appropriate words not to alter the original meaning, each keyword will trigger unlikely responses and can be easy to detect and over-written during fine-tuning. In case of rare keywords fine tuning is more complex because attacker does not have access to final weights and contains poisoned pre-trained weights. Attacker objective is to find a set of parameters that satisfies the loss function and achieves the target class. There are certain settings done while experimenting the model and fine-tuning the procedure. Like full data knowledge, domain shift, concrete attack methods.

If keywords are uncommon then they are to be appear less frequently in tuning the dataset, it is assumed that they will be modified with little fine-tuning and have embeddings close to zero, target class before applying to RIPPLe is called embedding surgery and combined method called restricted inner product poison learning with embedding surgery. An experimental setting is performed to validate if pre-trained models can be poisoned through three classification tasks – sentiment classification, toxicity detection and spam detection. With experiments label flip rate is calculated. It is observed that model has given a promised 50% LFR in all the cases and LFR does not poison the performance even batch size is increased. Ablation

and use of proper nouns as trigger words can reduce the poisoning of the datasets even for labelled and unlabeled.

To conclude this paper, declares that pre-trained models are poisoned such that they are exposed to backdoors when fine-tuned. Most effective method is RIPPLES and capable of creating backdoors with high success rate and modifying training dataset to hyper parameter settings. This work analyzes the pre-trained weights and reduces the backdoor attacks.

Word Frequency does not predict Grammatical Knowledge

https://aclanthology.org/2020.emnlp-main.331.pdf

The language models perform text prediction and downstream tasks such as question-answering, text classification and natural language reference. These models raise scientific questions about the knowledge acquired. In this paper, focus is on variation of grammatical knowledge that exists in a neural language model. It is mostly about systematic sources of variations in judgements. The accuracy is measured by making grammatical judgements involving different nouns. Then possible nouns are found and the properties of these nouns are paradoxically easy to learn to model and variations are observed.

The model is built in order to investigate the grammatical judgements and minimal pair of sentences that are different from each other in their acceptability due to difference in grammatical property. The most occurred combination has higher probability to grammatical sentence. There are 10 templates to generate minimal pairs. These tasks fall into two categories – subject verb agreement and reflexive anaphora. Example – The cat next to the boy jumps, *The cat next to the boy jump. The RA tasks measure if the language model understands the conditions on reflexive pronouns. And in the end comp task evaluates model to understand

reflexives and in the same clause as their antecedents. This model accuracy can also be measured on a particular target noun. On replacing the pair of model grammatical judgements on 500 minimal pairs are computed and averaged resulting in task performance.

Methods used to calculate task performance score, are sentence generation. Target nouns are drawn from noun list and each pair of task template has a target noun and 500 random sentences generated from word pair lists. For each noun-verb combination 4 versions are generated singular grammatical and ungrammatical, plural grammatical and ungrammatical. Sentence scoring are calculated for a particular sampled sentence where for each variant model computes a score and this score is the log probability of the string. The BERT transformer is used to perform SVA tasks and compute probability for each target noun, similar way RA tasks are used to compute probability of reflexive pronoun and resulting conditional probability is calculated to get overall sentence score. Noun scoring is calculated on each target noun with sample of 500 sentences. Word filtering and replacement is performed like replacing verb or reflexive pronoun with a masked token and each sentence will have a masked token corresponding to a word that should agree with target noun. The performance of nouns is evaluated within the model across different tasks and also across different models. The results have proven that noun exhibit stable task performance across language models and correlated with features of training data.

There are few nouns whose performance varies on grammatical tasks, so the variation Is not explained by frequency in natural text. Nouns that occur on order of 100 in corpus do not have systematically worse performance than nouns that occur 10^6 times. It is concluded that if low frequency nouns are understood as well as higher frequency nouns then this suggests that

language models few-shot learn the grammatical properties of nouns. So, experiments are conducted on frequency of nouns that are singular, plural and phrasal constructions perform better on reflexive anaphora tasks.

To conclude that, though there are sufficient samples to observe occurrence of a noun, these samples are missed in training the dataset causing noun to degrade. This explanation is lead to source of the problem, it will be more severe in case of infrequent nouns than the frequent nouns, training samples will have longer intervals between them for infrequent nouns, Thus would predict better performance of frequent nouns, but will be a failure in case of infrequent nouns.

I believe further training the model with better datasets like book reviews, can improve its accuracy and performance on infrequent nouns.

References:

- https://aip.scitation.org/doi/full/10.1063/5.0134317?gclid=Cj0KCQjwtsCgBhDEARIsAE7RYh0EwUhHWSygZm9QzX9_orJgqUx7blB3HmGSg2RuXt1hKXQ2JNfyzZMaAlQAEALw_wcB
- https://en.wikipedia.org/wiki/Neural_network
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9

- https://medium.com/sciforce/a-comprehensive-guide-to-natural-language-generation-dd63a4b6e548