IC ISDM

RMDL: Random Multimodel Deep Learning for Classification

Kamran Kowsari, Mojtaba Heidarysafa, Donald E. Brown,
Kiana Jafari Meimandi, and Laura E. Barnes

Outline

- Introduction
- Related Works and Baseline
  - Image Classification
  - Text Classification
- RMDL: Random Multimodel Deep Learning
  - Deep Neural Networks (DNN)
  - Recurrent Neural Networks (RNN)
  - Convolutional Neural Networks (CNN)
  - RMDL
- Results Experimental
  - Text Results
  - Image Results
- Conclusion

Introduction (Motivation)

- The exponential growth in the number datasets requires :
  - Robust and accurate data classification
- Deep learning approaches have achieved surpassing results, but
  - Finding the suitable structure and architecture for these models has been an important challenge
- Which deep learning approach is suitable?
  - Deep Neural Network (DNN)
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Convolutional Recurrent Neural Network (CRNN)
  - etc.
- Which deep learning Structure is suitable?
  - How many hidden layer?
  - How many nodes?
  - Which optimizer ?
  - etc.

Introduction (Motivation)

- RMDL (Random Multimodel Deep Learning)
  - Ensembles of deep learning architectures
  - Random
    - Randomly generate number of hidden layer
    - Randomly choose number node in each hidden layer
  - Multimodel
    - Generate *n* number of models
  - Deep Learning
    - Generate $r + c + d = n$. which
    - *d* random model of DNN classifiers,
    - *c* models of CNN classifiers,
    - *r* RNN classifiers

Introduction (Goal of Project)

- RMDL (Random Multimodel Deep Learning)
  - Which deep learning approach is suitable? (Deep Neural Network (DNN), Convolutional Neural Network CNN), Recurrent Neural Network (RNN))
  - Which deep learning Structure is suitable? (How many node and hidden layer)
  - Which optimizer ? (SGD, adam, and etc.)

Related Works (Text)

- Recurrent Neural Networks (RNN)
  - Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical Attention Networks for Document Classification," in *HLT-NAACL*, pp. 1480-1489.
- Convolutional Neural Networks (CNN)
  - Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*,
- Deep Neural Networks (DNN)
- Support Vector Machine (SVM)
  - K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from TF-IDF to TF-IGM for term weighting in text classification," (in English), *Expert Systems with Applications*, vol. 66, pp. 245-260, Dec 30 2016.
- Support Vector Machine (SVM)
  - A. Sun and E.-P. Lim, "Hierarchical text classification and evaluation," in *ICDM*, pp. 521-528.
- Naive Bayes Classification (NBC)
  - S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some effective techniques for naive bayes text classification," (in English), *IEEE transactions on knowledge and data engineering*, vol. 18, no. 11, pp. 1457-1466, Nov 2006.
- Hierarchical Deep Learning for Text Classification (HDLTex)
  - K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: Hierarchical Deep Learning for Text Classification," in *ICMLA*, 2017, pp. 364-371.

Related Works (Image)

- Deep L2-SVM
  - Y. Tang, "Deep learning using linear support vector machines," arXiv preprint arXiv:1306.0239, 2013.
- Maxout Network
  - I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," arXiv preprint arXiv:1302.4389, 2013.
- BinaryConnect
  - M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in Advances in Neural Information Processing Systems, pp. 3123-3131
- PCANet-1
  - T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?," IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 5017- 5032, Dec 2015.
- gcForest
  - Z.-H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," arXiv preprint arXiv:1702.08835, 2017.

Feature Extraction (tf-idf)

- Term Frequency-Inverse Document Frequency:

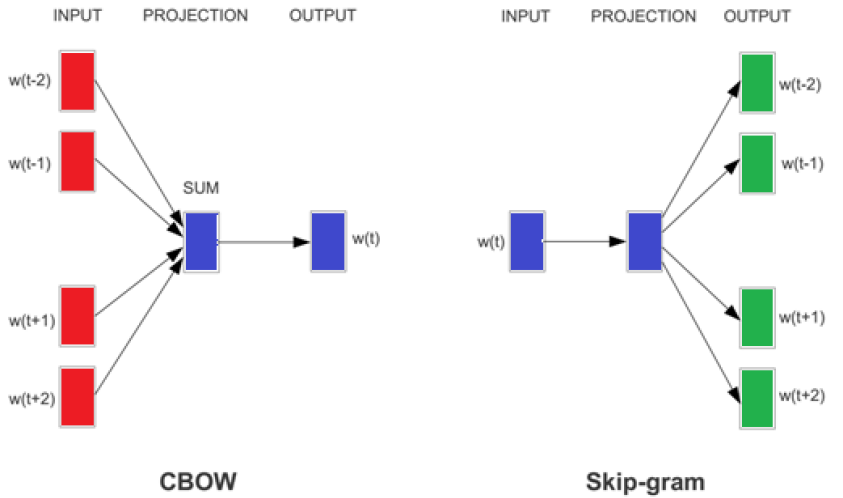
$$W(d,t) = TF(d,t) * log(\frac{N}{df(t)})$$

- Where N is number of documents and $df(t)$ is the number of documents containing the term *t* in the corpus.
- So we have a |V|-dimensional vector space
- Terms are axes of the space
- Documents are points or vectors in this space
- Very high-dimensional: tens of millions of dimensions.

Feature Extraction (Word2Vec)

- Vector Representation of Text
  - Word2Vec:
    - T. Mikolov et al. presented “word to vector” representation as a better word embedding architecture.
    - Word2vec approach uses two neural networks namely continuous bag of words (CBOW) and continuous skip-gram to create a high dimension vector for each word.
    - Represent each word with a low-dimensional vector
    - Word similarity = vector similarity

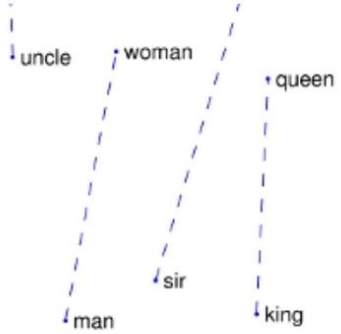
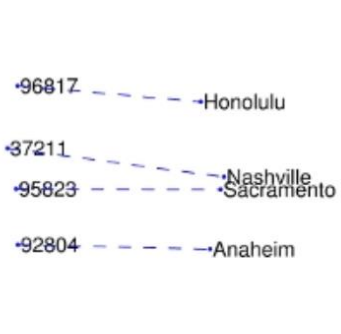
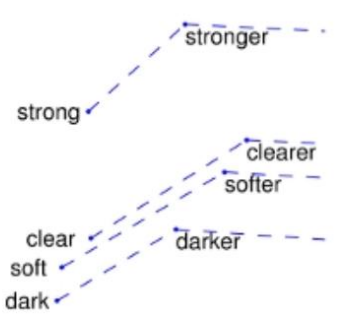
- basic neural network models:
  - Continuous Bag of Word (CBOW): use a window of word to predict the middle word
  - Skip-gram (SG): use a word to predict the surrounding ones in window.



Feature Extraction (GloVe)

- Global Vectors for Word Representation (GloVe):
  - The approach is very similar to "word to vector" method where each word is presented by a high dimension vector and trained based on the surrounding words over a huge corpus.

nearest neighbors of *frog*	Litoria	Leptodactylidae	Rana	Eleutherodactylus
Pictures				

Comparisons	man -> woman	city -> zip	comparative -> superlative
GloVe Geometry			

RMDL

- RMDL: Random Multimodel Deep Learning
  - Deep Neural Networks (DNN)
  - Recurrent Neural Networks (RNN)
  - Convolutional Neural Networks (CNN)

$$\hat{y}_{ij} = \begin{bmatrix} \hat{y}_{i1} \\ \vdots \\ \hat{y}_{ij} \\ \vdots \\ \hat{y}_{in} \end{bmatrix}$$

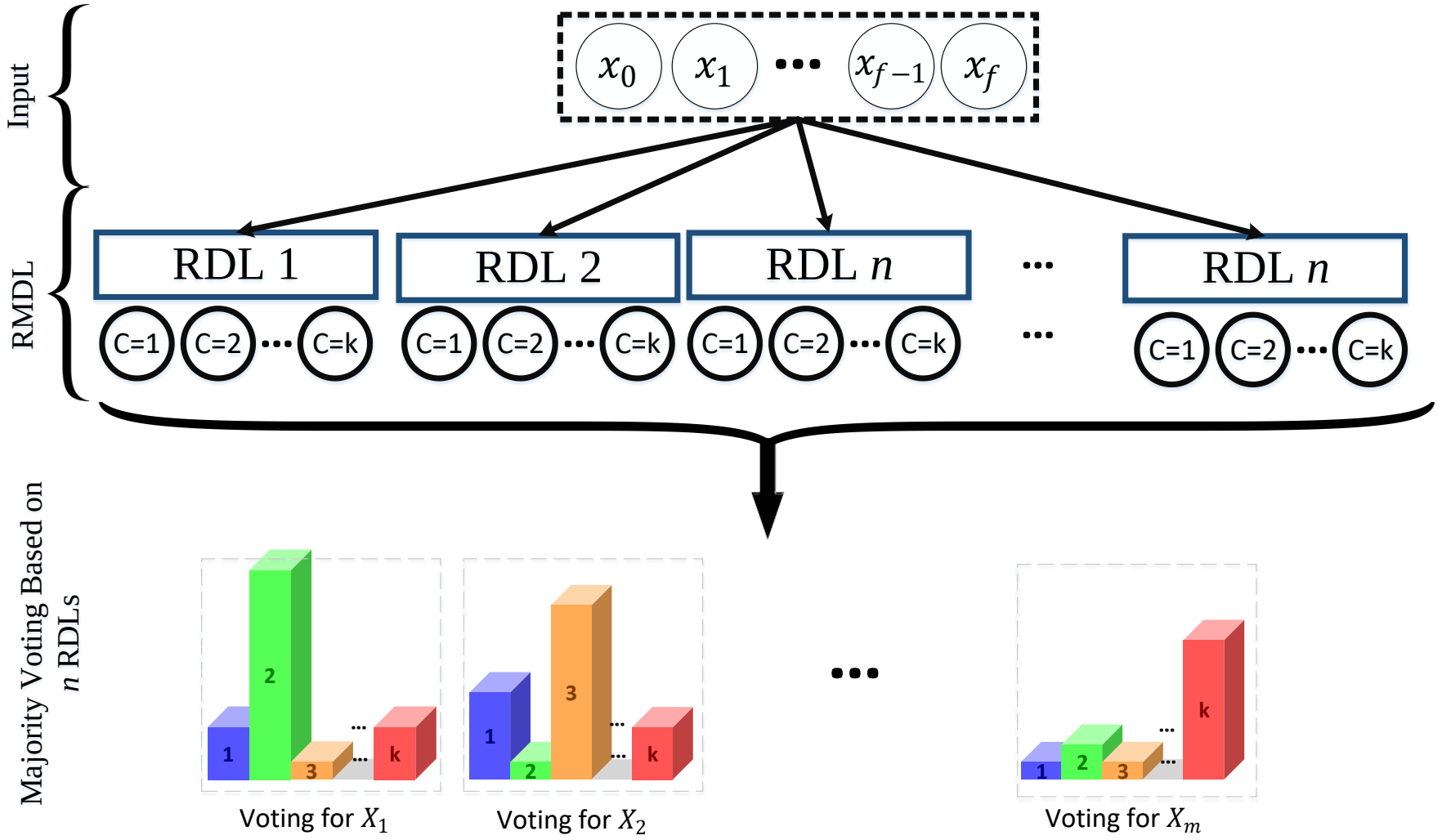
  - Where n is number of random model, and $\hat{y}_{ij}$ shows the prediction of label of document or data point of $D_i \in \{x_i, y_i\}$for model j and $\hat{y}_{ij}$ is defined as follows:

$$\hat{y}_{ij} = \arg\max_{k} [softmax(y_{ij})]$$

  - Then majority vote:

$$M(y_{i1}, y_{i2}, .. y_{in}) = \left\lfloor \frac{1}{2} + \frac{(\sum_{j=1}^{n} y_{ij}) - \frac{1}{2}}{n} \right\rfloor$$

RMDL (Random Multimodel Deep Learning)



RMDL (DNN)

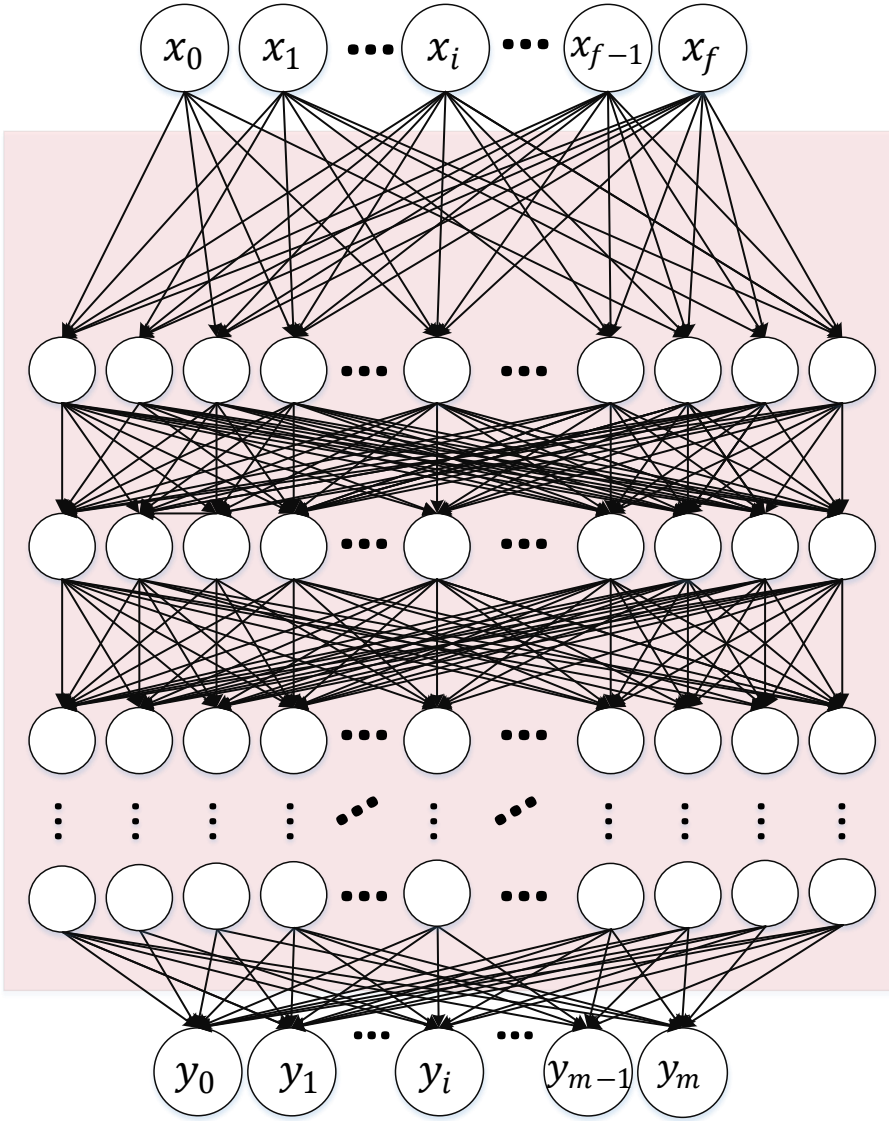
- Deep Neural Networks' structure is designed to learn by **multi connection** of layers (each layer only **receives connection from previous** and **provides connections only to the next layer**)
- The input is a connection of feature space with first hidden layer.
- Deep Neural Networks (DNN) is discriminative trained model that uses standard back-propagation algorithm using sigmoid and ReLU:

$$f(x) = \frac{1}{1 + e^{-x}} \in (0,1)$$

$$f(x) = \max(0, x)$$

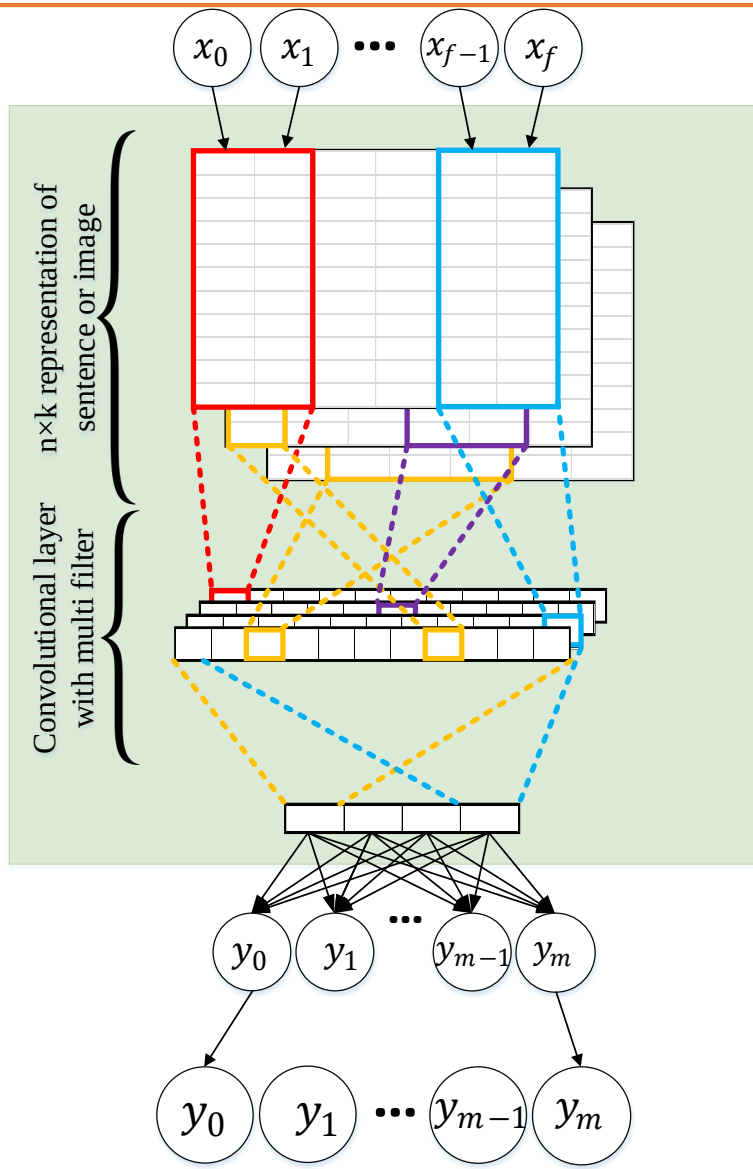
- For multi-class classification, should use Softmax:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad j \in \{1, \ldots, K\}$$



RMDL (CNN)

- CNN have been effectively used for text classification
- image processing an image tensor is convolved with a set of kernels of size $d \times d$.
- For Image 2D Convolutional Layers and 2D max pool are used.
- For Text 1D Convolutional Layers and 1D max pool are used.



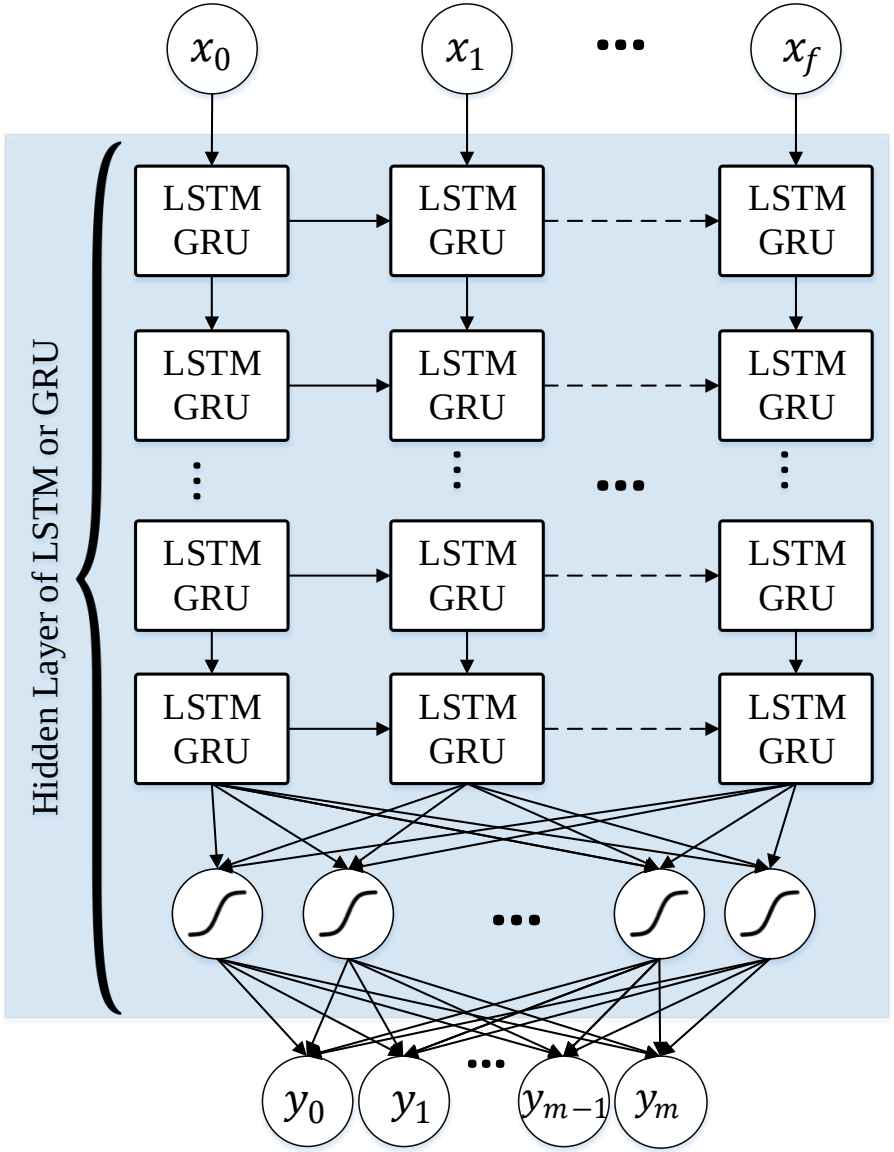
RMDL (RNN)

- Another neural network architecture that contributes in RMDL is Recurrent Neural Networks (RNN)
- RNN assigns more weights to the previous data points of sequence.
- In RNN the neural net considers the information of previous nodes in a very sophisticated method which allows for better semantic analysis of structures of dataset.
- General formulation of this concept is given:

$$x_t = F(x_{t-1}, \mathbf{u_t}, \theta)$$

- weights to formulate with specified parameters:

$$x_t = \mathbf{W_{rec}}\sigma(x_{t-1}) + \mathbf{W_{in}}\mathbf{u_t} + \mathbf{b}$$



RMDL (RNN)

Long Short-Term Memory (LSTM)

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i)$$

$$\tilde{C}_t = \tanh(W_c[x_t, h_{t-1}] + b_c)$$

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f)$$

$$C_t = i_t * \tilde{C}_t + f_t C_{t-1}$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o)$$

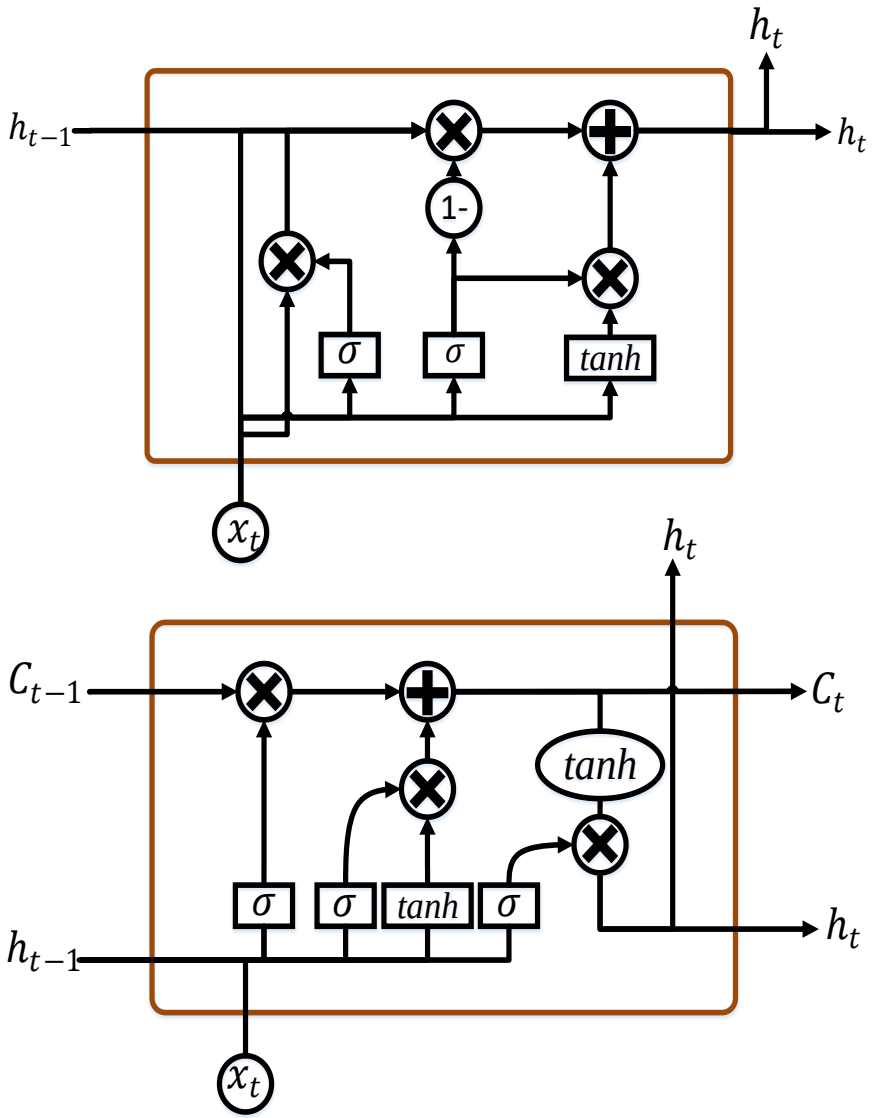
$$h_t = o_t \tanh(C_t)$$

Gated Recurrent Unit (GRU)

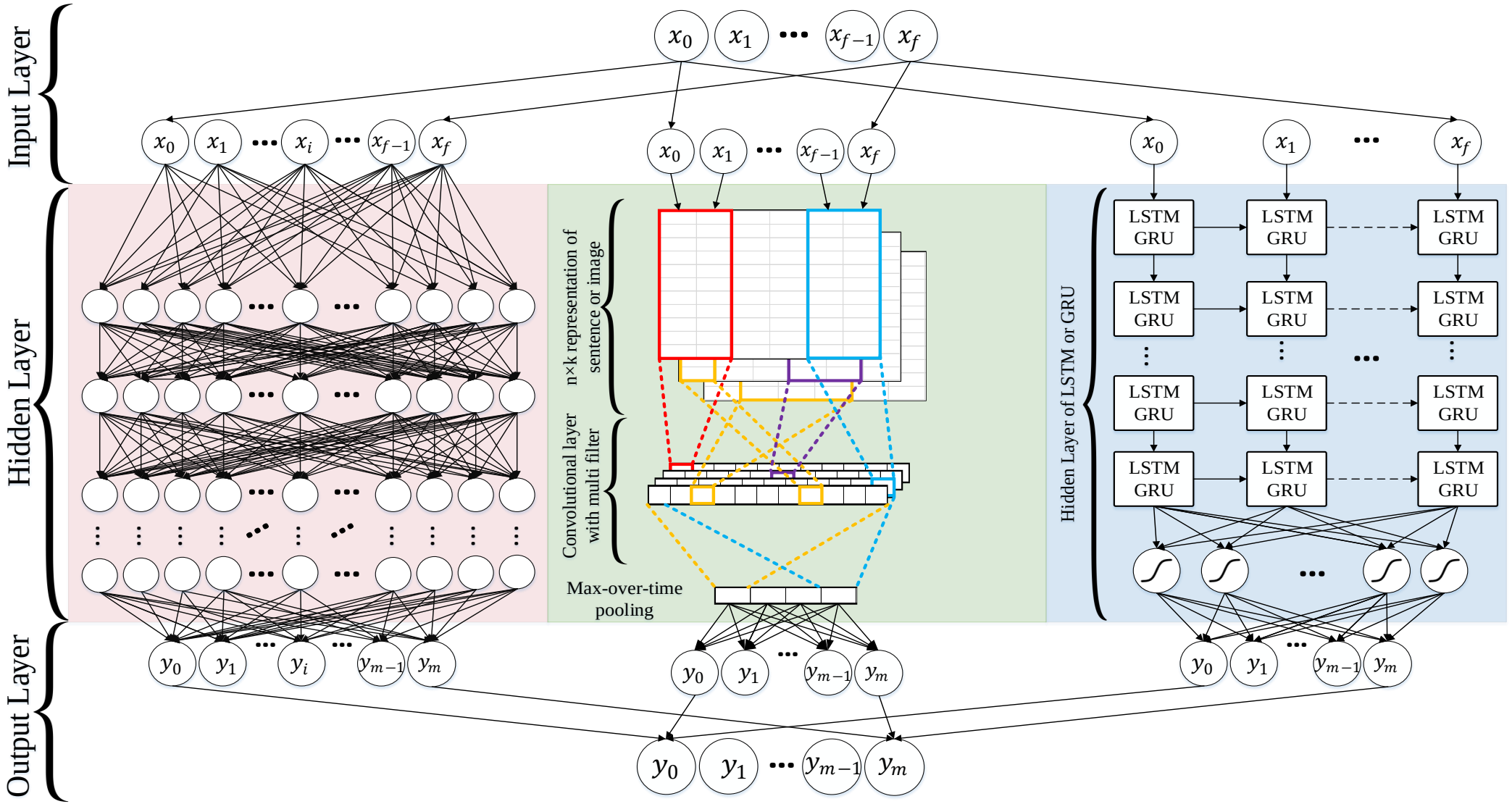
$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$

$$\tilde{r}_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r),$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)$$

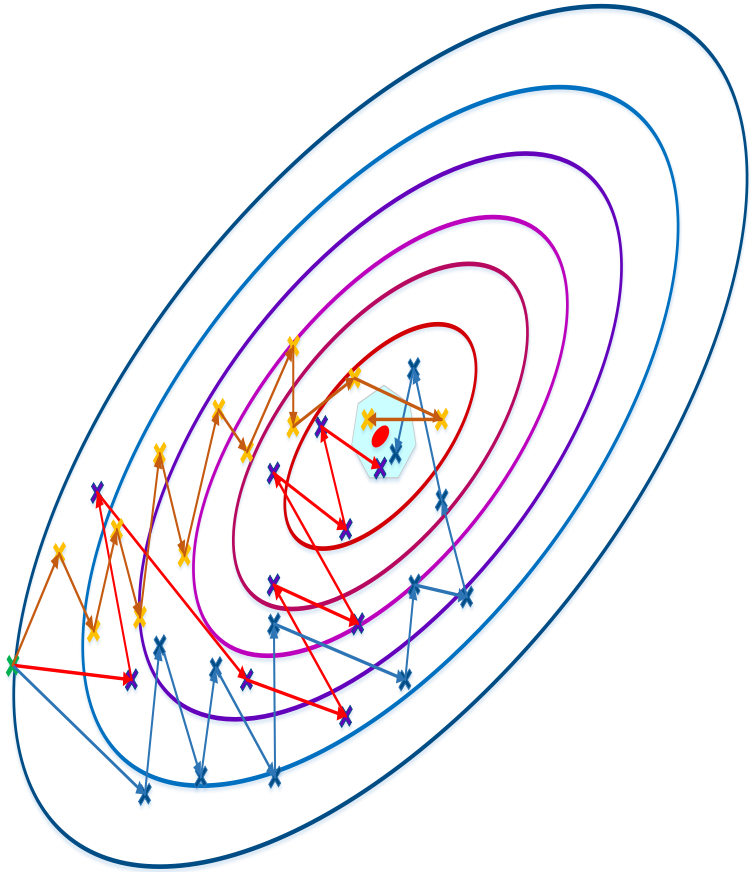


RMDL


Input Layer
Hidden Layer
Output Layer
n×k representation of sentence or image
Convolutional layer with multi filter
Max-over-time pooling
Hidden Layer of LSTM or GRU
LSTM GRU

RMDL (Optimization)

- In RMDL we used different techniques of optimizers:
  - Stochastic Gradient Descent (SGD)
  - RMSprop
  - Adam Optimizer
  - Adagrad Optimizer



RMDL (Optimization)

- In RMDL we used different techniques of optimizers:
  - Stochastic Gradient Descent (SGD)
    - The fundamental equation for Stochastic Gradient Descent (SGD) is shown:

$$\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta, x_i, y_i)$$

    - for updating parameters

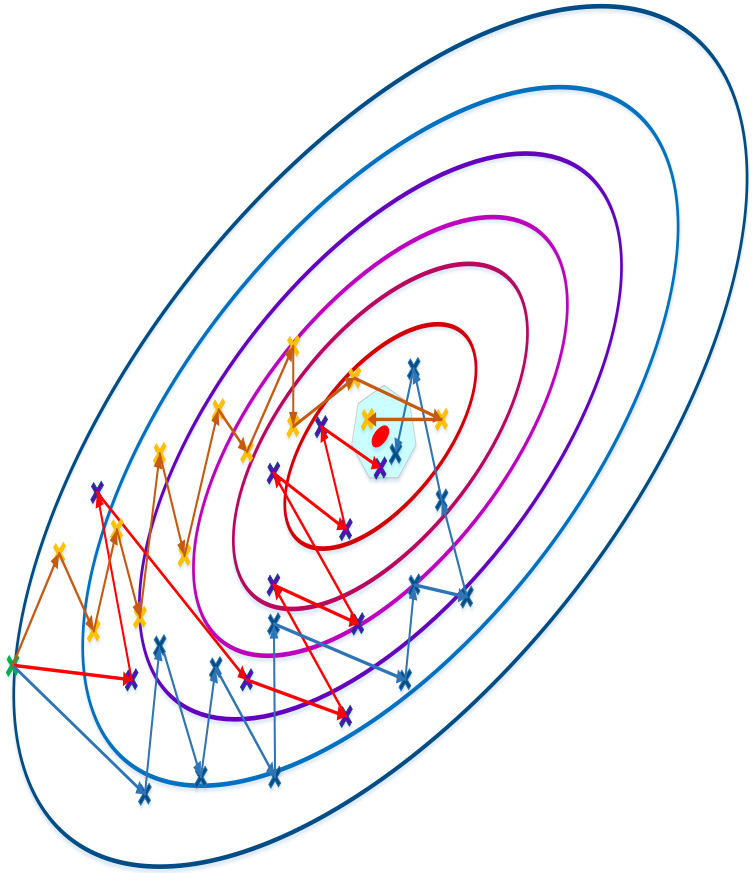
$$\theta \leftarrow \theta - (\gamma\theta + \alpha \nabla_\theta J(\theta, x_i, y_i))$$

  - RMSprop
    - divide the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight.
    - The equations of the momentum:

$$v(t) = \alpha\, v(t-1) - \epsilon \frac{\partial E}{\partial w}(t)$$

$$\Delta w(t) = v(t) = \alpha\, v(t-1) - \epsilon \frac{\partial E}{\partial w}(t) = \alpha\, \Delta v(t-1) - \epsilon \frac{\partial E}{\partial w}(t)$$

    - RMSProp does not do bias correction which will be a significant problem



RMDL (Optimization)

- In RMDL we used different techniques of optimizers:
  - Adam Optimizer
    - Adam is another stochastic gradient optimizer which uses only the first two moments of gradient.

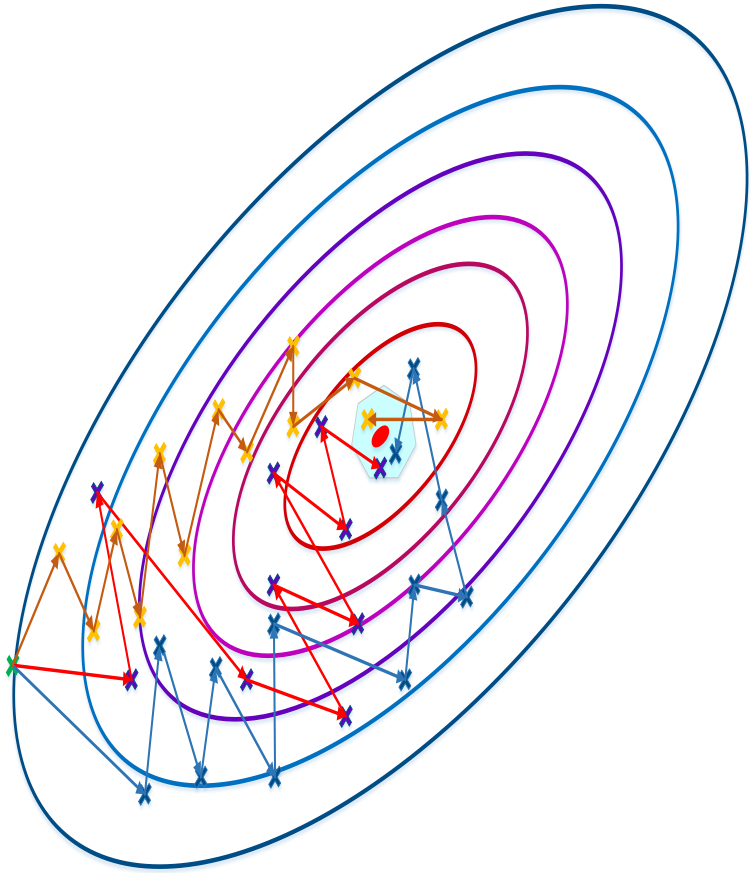
$$\theta \leftarrow \theta - \frac{\alpha}{\sqrt{\hat{v}} + \epsilon}\hat{m}$$

$$g_{i,t} = \nabla_\theta J(\theta_i, x_i, y_i)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_{i,t}$$

$$m_t = \beta_2 v_{t-1} + (1-\beta_2) g_{i,t}^2$$

    - Where m_t is the first moment and v_t indicates second moment that both are estimated. $\widehat{m_t} = \frac{m_t}{1-\beta_1^t}$ and $\widehat{v_t} = \frac{v_t}{1-\beta_2^t}$



RMDL (Optimization)

- In RMDL we used different techniques of optimizers:
  - Adagrad Optimizer
    - Adagrad is a novel family of sub-gradient methods which dynamically absorb knowledge of the geometry of the data to perform more informative gradient based learning.
    - AdaGrad is an extension of SGD. In iteration k, define:

$$G^{(k)} = diag\left[\sum_{i=1}^{k} g^{(i)}\left(g^{(i)}\right)^{T}\right]^{\frac{1}{2}}$$

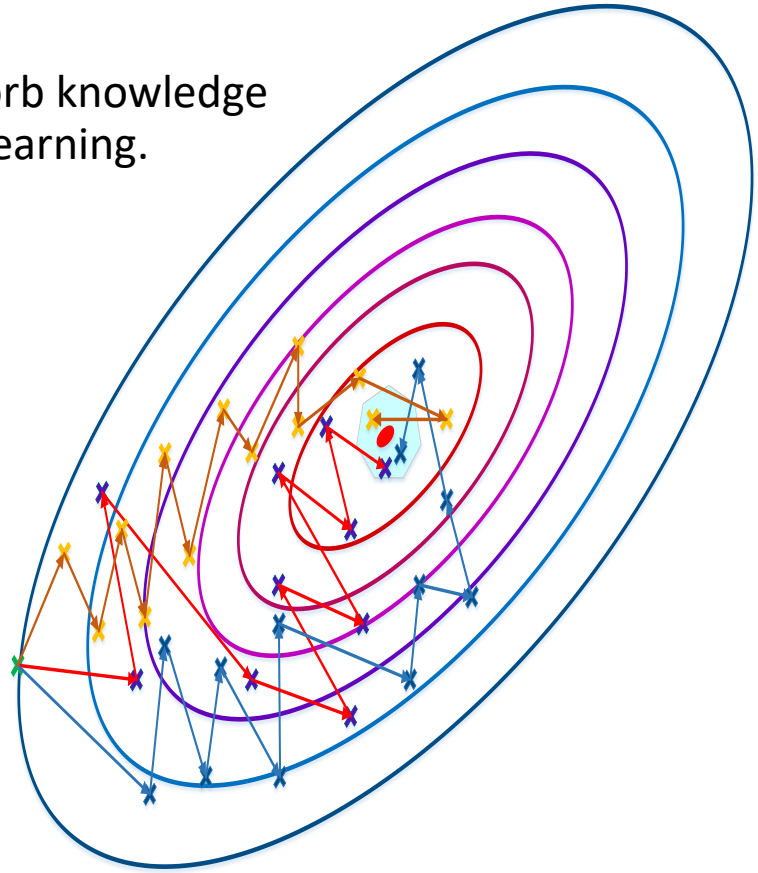
    - diagonal matrix:

$$G_{jj}^{(k)} = \sqrt{\sum_{i=1}^{k}(g_i^{(i)})^2}$$

    - Update

$$x^{(k+1)} = arg\min_{x\in X}\{\langle \nabla f(x^{(k)}), x\rangle + \frac{1}{2\alpha_k}\left|\left|x - x^{(k)}\right|\right|_{G^{(k)}}^2\} =$$

$$= x^{(k)} - \alpha B^{-1}\nabla f(x^{(k)}) \qquad (\text{if } X = R^n)$$



Experimental Results (Text)

- Dataset:
  - Reters-21578 Dataset
    - This dataset contains 21,578 documents with 90 categories.
  - Web of Science Dataset
    - Web of Science Dataset WOS-11967
      - This dataset contains 11,967 documents with 35 categories which include 7 parents categories.
    - Web of Science Dataset WOS-46985
      - This dataset contains 46,985 documents with 134 categories which include 7 parents categories.
    - Web of Science Dataset WOS-5736
      - This dataset contains 5,736 documents with 11 categories which include 3 parents categories.
  - 20Newsgroups Dataset
    - This dataset contains 20,000 documents with 20 categories.
  - IMDB Dataset
    - This dataset contains 50,000 documents with 2 categories.

Experimental Results (Text)

Methods		Dataset			
		WOS-5736	W-11967	WOS-46985	Reuters-21578
Baseline	DNN	86.15	80.02	66.95	85.3
	CNN	88.68	83.29	70.46	86.3
	RNN	89.46	83.96	72.12	88.4
	NBC	78.14	68.8	46.2	83.6
	SVM	85.54	80.65	67.56	86.9
	SVM (TF-IDF)	88.24	83.16	70.22	88.93
	Stacking SVM	85.68	79.45	71.81	NA
	HDLTex	90.42	86.07	76.58	NA
RMDL	**3 RDLs**	**90.86**	**87.39**	**78.39**	**89.10**
	9 RDLs	**92.60**	**90.65**	**81.92**	**90.36**
	15 RDLs	**92.66**	**91.01**	**81.86**	**89.91**
	30 RDLs	**93.57**	**91.59**	**82.42**	**90.69**

Experimental Results (Text)

Methods		Dataset	
		IMDB	20NewsGroup
Baseline	DNN	88.55	86.50
	CNN	87.44	82.91
	RNN	88.59	83.75
	Naive Bayes Classifier	83.19	81.67
	SVM	87.97	84.57
	SVM(TF-IDF)	88.45	86.00
RMDL	**3 RDLs**	**89.91**	**86.73**
	9 RDLs	**90.13**	**87.62**
	15 RDLs	**90.79**	**87.91**

Experimental Results (Image)

- Dataset:
  - MNIST Dataset
    - The MNIST database contains 60,000 training images and 10,000 testing images.
  - CIFAR-10 Dataset
    - The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
  - The Database of Faces (The Olivetti Faces Dataset)
    - he files are in PGM format, and can conveniently be viewed on UNIX (TM) systems using the 'xv' program. The size of each image is 92x112 pixels, with 256 grey levels per pixel. The images are organized in 40 directories (one for each subject), which have names of the form sX, where X indicates the subject number (between 1 and 40). In each of these directories, there are ten different images of that subject, which have names of the form Y.pgm, where Y is the image number for that subject (between 1 and 10).

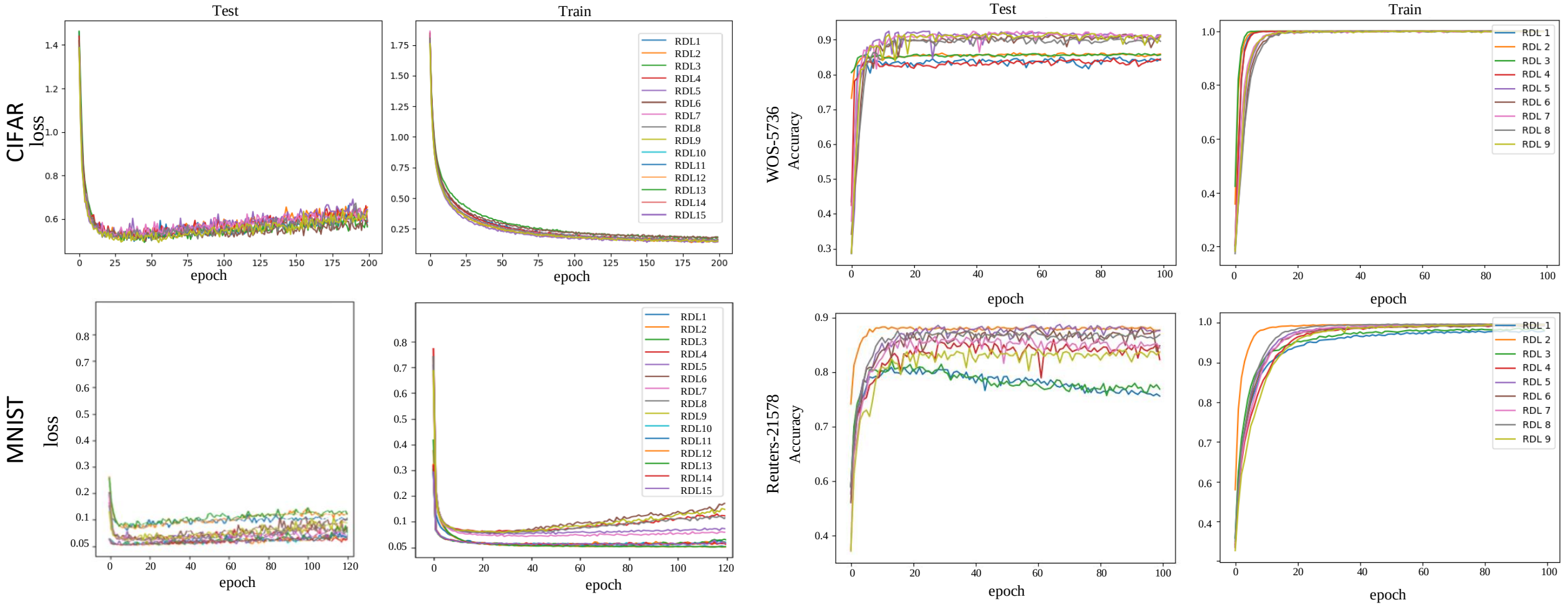
Experimental Results (Image)

Methods		MNIST	CIFAR-10
Baseline	Deep L2-SVM	0.87	11.9
	Maxout Network	0.94	11.68
	BinaryConnect	1.29	9.90
	PCANet-1	0.62	21.33
	gcForest	0.74	31.00
RMDL	**3 RDLs**	**0.51**	**9.89**
	9 RDLs	**0.41**	**9.1**
	15 RDLs	**0.21**	**8.74**
	30 RDLs	**0.18**	**8.79**

Experimental Results (Image)

Methods		5 Images	7 Images	9 Images
Baseline	gcForest	91.00	96.67	97.50
	Random Forest	91.00	93.33	95.00
	CNN	86.50	91.67	95.00
	SVM (rbf kernel)	80.50	82.50	85.00
	kNN	76.00	83.33	92.50
	DNN	85.50	90.84	92.5
RMDL	**3 RDL**	**93.50**	**96.67**	**97.5**
	9 RDL	**93.50**	**98.34**	**97.5**
	15 RDL	**94.50**	**96.67**	**97.5**
	30 RDL	**95.00**	**98.34**	**100.00**

Experimental Results (Epochs level)



Conclusion

- This paper presents a new technique using stat-of-art machine learning methods, deep learning.
- this paper introduces RMDL (Random Multimodel Deep Learning) for the classification that combines multi deep learning models to produce better performance, and solve following problems:
  - choosing the best structures of Deep Learning
  - choosing the best architectures of Deep Learning
- We've Evaluated this approach on datasets such as the Web of Science (WOS), Reuters, MNIST, CIFAR, IMDB, and 20NewsGroups.
- Our results show that such multi model deep learning structure can improve classification task on broad range of datasets by using majority vote.

Conclusion

- External Link :
  - Source code is shared at https://github.com/kk7nc/RMDL
  - Source code passed travis-ci at https://travis-ci.org/kk7nc/RMDL
  - Run Code:
    - There are git RMDL in this repository; to clone all the needed files, please use:

```
git clone --recursive https://github.com/kk7nc/RMDL.git
```

    - The primary requirements for this package are Python 3 with Tensorflow. The requirements.txt file contains a listing of the required Python packages; to install all requirements, run the following:

```
pip -r install requirements.txt
```

    - If the above command does not work, use the following:

```
pip3  install -r requirements.txt
```

    - Then you can run any of examples:

```
cd Examples
python MNIST.py
```

RMDL: Random Multimodel Deep Learning for Classification

Contributor

Kamran Kowsari
(kk7nc@virginia.edu)

Mojtaba Heidarysafa
(mh4pk@virginia.edu)

Donald E. Brown
(deb@virginia.edu)

Kiana Jafari Meimandi
(kj6vd@virginia.edu)

Laura E. Barnes
(lb3dp@virginia.edu)